

NO-A184 642

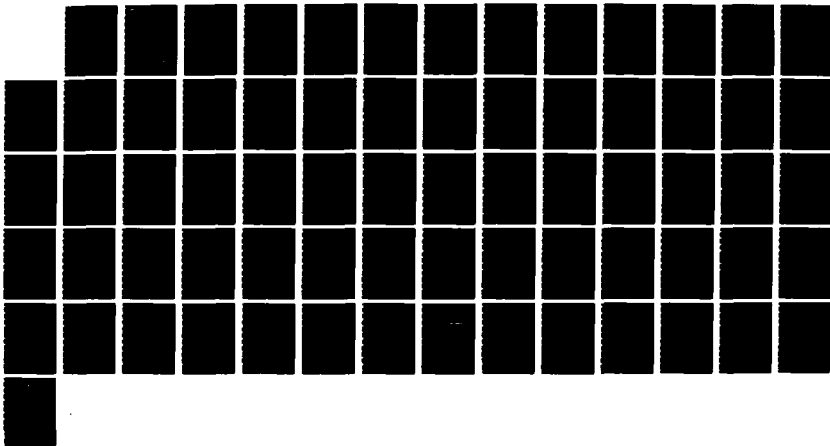
PROCEEDINGS OF WORKSHOP ON EXPERT SYSTEMS FOR  
CONSTRUCTION SCHEDULING HEL (U) CONSTRUCTION  
ENGINEERING RESEARCH LAB (ARMY) CHAMPAIGN IL

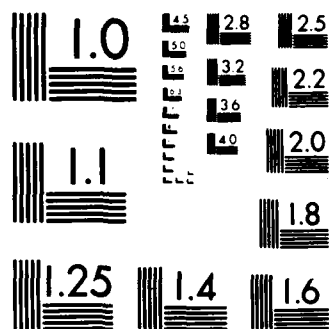
1/1

UNCLASSIFIED

M J O'CONNER ET AL AUG 87 CERL-CP-P-87/13 F/G 13/2

NL





MICROCOPY RESOLUTION TEST CHART  
NATIONAL BUREAU OF STANDARDS 1963 A



**US Army Corps  
of Engineers**  
Construction Engineering  
Research Laboratory

USA-CERL CONFERENCE PROCEEDINGS P-87/13  
August 1987

2

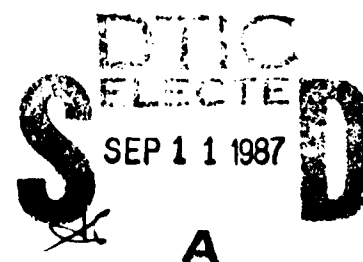
**AD-A184 642**

# Proceedings: Workshop on Expert Systems for Construction Scheduling

Edited by  
Michael J. O'Connor  
Jesus M. De La Garza

The U.S. Army Construction Engineering Research Laboratory sponsored a workshop on Expert Systems for Construction Scheduling during March 23-24, 1987 in Champaign, Illinois.

The goal of this workshop was to provide an opportunity for researchers in the area to communicate results of their work, exchange ideas, disseminate solutions to problems, and discuss the future of this topic area. Relevant topics of discussion included: automated network generation, network measurement, and network diagnostics.



Approved for public release; distribution is unlimited.

87 9 11 021

The contents of this report are not to be used for advertising, publication, or promotional purposes. Citation of trade names does not constitute an official indorsement or approval of the use of such commercial products. The findings of this report are not to be construed as an official Department of the Army position, unless so designated by other authorized documents.

*DESTROY THIS REPORT WHEN IT IS NO LONGER NEEDED  
DO NOT RETURN IT TO THE ORIGINATOR*

AD-A184642

REPORT DOCUMENTATION PAGE				Form Approved OMB No 0704 0188 Exp Date Jun 30 1986	
1a REPORT SECURITY CLASSIFICATION UNCLASSIFIED			1b RESTRICTIVE MARKINGS		
2a SECURITY CLASSIFICATION AUTHORITY			3 DISTRIBUTION AVAILABILITY OF REPORT Approved for public release; distribution is unlimited.		
2b DECLASSIFICATION/DOWNGRADING SCHEDULE					
4 PERFORMING ORGANIZATION REPORT NUMBER(S) USA-CERL CP P-87/13			5 MONITORING ORGANIZATION REPORT NUMBER(S)		
6a NAME OF PERFORMING ORGANIZATION U.S. Army Construction Engr Research Laboratory		6b OFFICE SYMBOL (if applicable) USA-CERL		7a NAME OF MONITORING ORGANIZATION	
6c ADDRESS (City, State, and ZIP Code) P.O. Box 4005 Champaign, IL 61820			7b ADDRESS (City, State, and ZIP Code)		
8a NAME OF FUNDING SPONSORING ORGANIZATION HQUSACE		8b OFFICE SYMBOL (if applicable)		9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER	
8c ADDRESS (City, State, and ZIP Code) 20 Massachusetts Ave., N.W. Washington, D.C. 20314-1000			10 SOURCE OF FUNDING NUMBERS		
			PROGRAM ELEMENT NO 4A161102	PROJECT NO AT23	TASK NO A
			WORK UNIT ACCESSION NO 046		
11 TITLE (Include Security Classification) Proceedings: Workshop on Expert Systems for Construction Scheduling (Unclassified)					
12 PERSONAL AUTHOR(S) Edited by O'Connor, Michael J., De La Garza, Jesus M.					
13a TYPE OF REPORT Final		13b TIME COVERED FROM _____ TO _____		14 DATE OF REPORT (Year, Month, Day) 8/87	
15 PAGE COUNT 68					
16 SUPPLEMENTARY NOTES Copies are available from the National Technical Information Service Springfield, VA 22161					
17 COSATI CODES			18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)		
FIELD	GROUP	SUB GROUP	Expert systems workshops		
13	13		construction		
			scheduling		
19 ABSTRACT (Continue on reverse if necessary and identify by block number)					
<p>The U.S. Army Construction Engineering Research Laboratory sponsored a workshop on Expert Systems for Construction Scheduling during March 23-24, 1987, in Champaign, Illinois.</p> <p>The goal of this workshop was to provide an opportunity for researchers in the area to communicate results of their work, exchange ideas, disseminate solutions to problems, and discuss the future of this topic area. Relevant topics of discussion included: automated network generation, network measurement, and network diagnostics.</p>					
20 DISTRIBUTION AVAILABILITY OF ABSTRACT <input type="checkbox"/> UNCLASSIFIED/LIMITED <input checked="" type="checkbox"/> SAME AS RPT <input type="checkbox"/> DTIC USERS			21 ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED		
22a NAME OF RESPONSIBLE INDIVIDUAL D. P. Mann			22b TELEPHONE (Include Area Code) (217) 373-7223		22c OFFICE SYMBOL Chief-IMT

## CONTENTS

	Page
DD 1473	1
FOREWORD	3
A Knowledge Engineering Approach to the Analysis and Evaluation of Construction Schedules for Vertical Construction (Michael J. O'Connor, C. William Ibbs, and Jesus M. De La Garza) .....	7
Expert Systems Development at Stone and Webster Engineering Corporation (Kenneth F. Reinschmidt) .....	11
Expert Systems for Construction Scheduling— Research at Carnegie Mellon University ; .....	14
(Chris Hendrickson and Daniel Rehak)	
Automated Planning Tool--A Testbed for Knowledge-Based Project Management (Glenn M. Yoshimoto) .....	17
Construction Knowledge Systems: Status of Work at the University of Texas (David B. Ashley) .....	21
Research at M.I.T. on Application of Knowledge Based Systems to the Project Control Process (Robert D. Logcher) .....	25
Adding Knowledge Based Systems Technology to Project Control Systems (Robert D. Logcher) .....	30
Thrust for Research Program in Computation for the Center for Advanced Construction Technology (Robert D. Logcher) .....	42
CALLISTO: An Intelligent System for Supporting Project Management (Steven F. Roth) .....	51
Research on the Use of Artificial Intelligence Techniques to Support Project Management (Raymond E. Levitt and Catherine Perman) .....	58

## FOREWORD

These proceedings were printed for Headquarters, U.S. Army Corps of Engineers (HQUSACE) by the Facility Systems Division (FS), U.S. Army Construction Engineering Research Laboratory (USA-CERL). The work was performed under Project 4A161102AT23, "Basic Research in Military Construction"; Task A, "A Base/Facility Development"; Work Unit 046, "A Physical Process Visualization Technique for Generating Networks."

The conference was organized by Mssrs. Michael J. O'Connor and Jesus M. De La Garza (USA-CERL-FS), P.O. Box 4005, Champaign, Illinois 61820-1305, (217) 373-7267 and (217) 352-6511, ext. 651, respectively.

Mr. E. A. Lotz is Chief of USA-CERL-FS. COL Norman C. Hintz is Commander and Director of USA-CERL and Dr. L. R. Shaffer is Technical Director.



APPROVED	
DATE	
BY	
FOR	
REVIEWED	
DATE	
BY	
FOR	
Dist	
ft-1	

## CONTENTS

	Page
DD 1473	1
FOREWORD	3
A Knowledge Engineering Approach to the Analysis and Evaluation of Construction Schedules for Vertical Construction (Michael J. O'Connor, C. William Ibbs, and Jesus M. De La Garza) .....	7
Expert Systems Development at Stone and Webster Engineering Corporation (Kenneth F. Reinschmidt) .....	11
Expert Systems for Construction Scheduling-- Research at Carnegie Mellon University .....	14
(Chris Hendrickson and Daniel Rehak)	
Automated Planning Tool--A Testbed for Knowledge-Based Project Management (Glenn M. Yoshimoto) .....	17
Construction Knowledge Systems: Status of Work at the University of Texas (David B. Ashley) .....	21
Research at M.I.T. on Application of Knowledge Based Systems to the Project Control Process (Robert D. Logcher) .....	25
Adding Knowledge Based Systems Technology to Project Control Systems (Robert D. Logcher) .....	30
Thrust for Research Program in Computation for the Center for Advanced Construction Technology (Robert D. Logcher) .....	42
CALLISTO: An Intelligent System for Supporting Project Management (Steven F. Roth) .....	51
Research on the Use of Artificial Intelligence Techniques to Support Project Management (Raymond E. Levitt and Catherine Perman) .....	58

## CONTENTS

	Page
Construction Scheduling: Expert Systems Development (Romey Ross) .....	62
Construction Scheduling Issues: A Construction Firm's Perspective (Paul M. Teicholz) .....	65
DISTRIBUTION	

**U.S. Army Construction Engineering Research Laboratory  
Workshop on Expert Systems for Construction Scheduling  
March 23-24, 1987, Champaign, Illinois**

**A Knowledge Engineering Approach to the Analysis and  
Evaluation of Construction Schedules  
for Vertical Construction**

**Michael J. O'Connor, C. William Ibbs, and Jesus M. De La Garza**

The U.S. Army Construction Engineering Research Laboratory (USA-CERL) and the University of Illinois Construction Engineering Expert Systems Laboratory (CEESL) have been working together to develop a knowledge-based system for analysis of construction schedules.

The primary research objectives are to extract, formalize, and articulate (1) empirical and judgmental knowledge about construction schedule analysis and (2) traditional project management theory to develop a prototype knowledge-based system. This system will assist field engineers in analyzing and modifying construction schedules of medium-rise to high-rise reinforced concrete buildings. Scheduling analysis and evaluation was divided into two areas, namely an Initial Schedule analysis module and an In-Progress Schedule analysis module. Each was based upon four major subcategories: (a) cost; (b) time; (c) logic; and (d) general requirements. The Initial Schedule module analyzes the initial planning schedule that contractors provide owners for verification at the outset of the project. Project managers need answers to questions like: What is the overall degree of schedule criticality?, etc. The In-Progress Schedule evaluation module allows project managers to investigate delay and duration modification concerns. For example, project managers seek answers to questions like: Are winter sensitive activities scheduled during winter?, etc.

In order to accomplish these objectives, the following tasks were established: 1) **Knowledge acquisition:** Determine the scope and complexity of the task; Identify the domain experts; Select the benchmark construction schedule; Acquire knowledge; and Produce a "paper" knowledge base. 2) **Knowledge organization:** Identify and capture expressions of similar form that reappear frequently in the "paper" knowledge base. 3) **Knowledge representation:** Determine the specific target inference engine; Decide how the "paper" knowledge base should be represented in the inference engine; and Develop a mapping technique to translate the concepts, facts and rules into the corresponding inference engine syntax. 4) **Knowledge implementation:** Replace the "paper" knowledge base with an "electronic" knowledge base. 5) **Knowledge validation:** Evaluate the prototype system against case studies and define its boundaries.

Efforts at USA-CERL have been concentrated on applying PC-based expert systems technology to this problem domain. This work focused on building an add-on system to existing project management system software. The system is fully integrated by linking together database, project management, and expert systems technology on a single personal computer. This implementation takes advantage of the electronic databases generated by project management systems. Most of these

lower-level shells, however, are built upon a rule-based representation scheme only. The "if...then" based systems were found to be highly limited in terms of knowledge representation and use of knowledge. Moreover, linkages to an intermediary relational database manager had to be built to enable communication between the existing project management system software and the expert system shell.

Concurrent efforts at the University of Illinois have focused upon applying a higher level programming environment system which allows more flexibility of knowledge representation and manipulation. The Automated Reasoning Tool (ART)<sup>TM</sup> programming environment has been selected and acquired as the inference engine to process the knowledge base. The knowledge architecture schemes of semantic nets, frames and object-oriented programming have provided drastic improvements in the representation of heuristic information.

The first exploratory research step was to determine the breath and depth of the construction schedule analysis domain. This step defined whether the Initial and In-Progress schedule analyses, as defined herein, were sufficiently well defined and self-contained. The aim is not for a system that is intricately tied to other kinds of knowledge, e.g., automated schedule generation. Rather, the goal is to develop a system that is expert in a limited, yet functional problem domain.

The sources of construction schedule expertise utilized thus far can be categorized into three groups: a) contractors; b) owners; and c) in-house. W.E. O'Neil and Pepper Construction companies, large building contractors in Chicago, have collaborated on this knowledge engineering project by designating one senior project manager who has committed the necessary time to the development of the system. Representatives from USA-CERL articulated an owner's view. Finally, the in-house expertise of several faculty members in the Civil Engineering Dept. has been drawn upon to contribute to the refinement and extension of both contractors' and owner's view.

A "paper" knowledge base consisting of English statements, which expressed the facts, concepts, and rules that the USA-CERL experts provided, was produced first. By showing this "paper" knowledge base to the other experts early in the project, it was possible to obtain a better understanding of the different kinds of expertise prevalent in the domain and which expert practiced which kinds. In addition, the senior project managers better understood the scope and complexity of this project. In all truthfulness, getting these experts to concentrate strictly on a narrow aspect of the problem has not been easy.

At this stage of the knowledge acquisition process, a wholesale effort began to acquire knowledge and to identify the kinds of problem-dependent strategies the contractors use. Two main techniques are being utilized to elicit the experts knowledge: 1) experts gave an account of their expertise by describing how they go about evaluating the "goodness" of a construction network; and 2) experts exercise their expertise in real problems, and then a model replicating their approach is generated.

As the "paper" knowledge base grew, it began to exhibit some regularity in the sense that expressions of similar form reappeared frequently. Once these regularities were identified, they were captured by building an English-like knowledge acquisition grammar. This grammar allowed expression of facts, rules, and concepts of the

construction schedule analysis domain. Use of this English-like knowledge acquisition grammar reduces the effort expended on acquiring additional rules. In addition, the knowledge represented in this generic syntax can be easily adapted to a variety of inference engine designs.

A mapping technique tailored to meet ART's specifications has been defined. This mapping technique relates the English-like knowledge acquisition grammar with ART's knowledge representation language. A different mapping technique can be designed for different inference engine, e.g., ART, KEE, Knowledge Craft (other proprietary, trademarked systems). However the result of this research will be useful and available to any interested party working in a system other than ART because this "paper" knowledge base will be readily transferable to other environments.

The development of the PC-based prototype has demonstrated that this new approach is satisfactory for accelerating many of the brute-force analyses and calculations typical of routine scheduling. However, this methodology cannot be shown to be a sufficient solution through the development of the prototype alone. Thus, subsequent experimentation and analyses are necessary to accomplish this.

Since formalizing and structuring the knowledge is more valuable than inference strategies, a major effort is being devoted to the expansion and refinement of the current knowledge base. Towards this end, an experiment is being designed with two video cameras, a trio of senior project managers, a rookie project manager, and a blue velvet curtain. The aim: to mimic a computer by having the trio act as the expert system, the curtain act as the computer screen and the rookie act as the user.

The USA-CERL and CEESL long-term research programs call for the development of a series of cohesive knowledge-based systems dedicated to: schedule, cost, quality and overhead control, and cost estimation for vertical construction. It is unrealistic to believe that one can build the complete system for scheduling control without a) eventual attachments to other elements of project control, and b) continued refinement, enhancement and updating.

## REFERENCES FOR ADDITIONAL INFORMATION REGARDING THIS RESEARCH

1. O'Connor, Michael J., Jesus M. De La Garza, and C. William Ibbs, "An Expert System for Construction Schedule Analysis," in *Expert Systems in Civil Engineering*, edited by Celal Kostem and Mary L. Maher, ASCE, pp. 67-77, Seattle, WA., April, 1986.
2. De La Garza, Jesus M., "A Knowledge Engineering Approach to the Analysis and Evaluation of Construction Schedules for Vertical Construction," Unpublished Ph.D. Thesis Proposal, University of Illinois at Urbana-Champaign, Dept. of Civil Engineering, June 1986.
3. De La Garza, Jesus M., and C. William Ibbs, "A Knowledge Engineering Approach to the Analysis and Evaluation of Vertical Construction Schedules," *Proceedings of the 10<sup>th</sup> Triennial Congress of the International Council for Building Research, Studies and Documentation, CIB.86 Congress -- Advancing Building Technology, Vol. 2, pp. 683-691, Washington, D.C., September, 1986.*
4. Ibbs, C. William, and Jesus M. De La Garza, "Knowledge Engineering for a Construction Scheduling Analysis System," *Expert Systems in Construction and Structural Engineering*, Edited by Hojjat Adeli, Chapman and Hall Publishing Co. forthcoming, 1987.
5. De La Garza, Jesus M., and C. William Ibbs, "Issues in Construction Scheduling Knowledge Representation," *Proceedings of the CIB W-65 Symposium -- Organization and Management of Construction, London, U.K., forthcoming, 1987.*

EXPERT SYSTEMS DEVELOPMENT AT  
STONE & WEBSTER ENGINEERING CORPORATION

Kenneth F. Reinschmidt

U.S. Army Construction Engineering Research Laboratory  
Workshop on Expert Systems for Construction Scheduling

Champaign, Illinois  
March 24, 1987

In 1983, the Stone & Webster Engineering Corporation established a group to investigate potential commercial applications of expert systems in engineering, design, construction, project management, and facilities management. In order to achieve client acceptance, the initial decision was made to focus on the delivery of expert systems, using the installed base of computer hardware available at construction sites. Consequently, the decision was made to use IBM PCs as the expert system delivery platforms.

The approach used at Stone & Webster was that, to the maximum extent possible, domain experts should develop expert systems themselves with only advice and guidance from knowledge engineers. Programming was to be confined to standard engineering languages (Fortran, Basic, etc.), database query languages (SQL), and graphics interfaces. No development was to be done in LISP or Prolog. It was believed that the commercial software industry would provide expert systems shells for PCs, but at that time no satisfactory PC shells were available.

Therefore, in 1984 Stone & Webster wrote its own PC expert system shell, Microcomputer Artificial Intelligence Diagnostic Service, and the first applications, such as PumpPro for diagnosis of centrifugal pump problems, were made using this shell. In 1985 and 1986, numerous PC expert system shells were placed on the market and Stone & Webster evaluated many of them. At the present time, several such shells for PCs and ATs are in use, depending on the needs of the particular application, preference for forward or backward chaining, etc. In addition to the IBM PCs, expert system applications have been developed on DEC VAXs using OPS5 as well as commercial shells.

In the project management field, two areas for expert system development were identified: project planning and generation of feasible project schedules, and project monitoring and diagnosis of project progress. In 1985, development commenced on PC-based expert systems in these areas, by an experienced knowledge engineer and a project manager with considerable background in the development of project management systems. The expert system for project planning was intended to develop a project network by an interactive dialog with the user.

After some investigation, it was concluded that this approach to project planning and network generation would not be successful. The intended approach might work for standardized projects, but the projects built by Stone & Webster are generally quite different, with considerable variation due to the type of project, client requirements, site conditions, etc. It was therefore concluded that a graphical approach was needed, in which the construction planner could visualize the project construction plan in a more realistic way than with the conventional network. Rather than generating the project plan directly, the expert system should assist the project planner by computing material, labor, and equipment requirements for each contemplated work package, and recommending improvements to the plan that would level manpower requirements, shorten the duration, improve efficiency, etc.

Accordingly, the development of expert systems for construction planning was shifted from the microcomputer to the IBM mainframe. The reasons for this were the following:

The expert system would have access to the relational database management system DB2, which manages the Stone & Webster integrated project database and has access to all project data.

The expert system would have access to the computer graphics systems CATIA and CADAM, which are used at Stone & Webster to design the project and which contain the geometrical description of the entire facility.

The expert system could be accessed by a number of terminals, including IBM 5080 graphics workstations and IBM 3270/PC management workstations.

With this approach, the expert system for construction planning has direct access to all project data in the database. It also has direct access to the three-dimensional computer design models of the facility. And, as the planning of major projects is a team function, rather than a single-man operation, several participants can use it from different terminals.

The expert system for project planning is intended to function approximately as follows:

The project engineers and designers create the computer model of the complete facility in three dimensions. This is the design process now in use by Stone & Webster. In this 3-D design process, all interferences are eliminated.

The construction specialists review the three-dimensional design model for constructibility, access for equipment, and other factors. If problems are uncovered, they are resolved between the construction specialists and the appropriate designers.

The construction planner breaks down the complete facility model into a set of steps. Each step corresponds to a potential construction work package. Each step is a three-dimensional computer model representing the components erected in that work package. This is the proposed construction sequence model.

For each proposed work package, the computer graphics package computes the lengths, areas, and volumes of the three-dimensional components in that package. The results are placed in the relational database.

The expert system evaluates the proposed construction sequence for feasibility and access, identifies problem areas, and makes recommendations to the construction planner as required.

For each proposed work package, the expert system uses the appropriate factors from the database to translate the computed areas and volumes into yards of concrete, square feet of formwork, tons of reinforcing steel, tons of structural steel, and other relevant construction material quantities.

For each proposed work package, the expert system uses the component sizes, weights, and other parameters to determine construction equipment requirements and compares these requirements to project equipment availability.

For each proposed work package, the expert system selects the appropriate unit rate factors from the database to translate the computed material quantities into manhours for each labor category.

The expert system compares the derived manloading for the proposed construction sequence with total project manpower availability, identifies potential problem areas, and makes recommendations to the construction planner as required to improve the manloading.

This process continues iteratively, with the construction planner and the expert system interacting until a satisfactory construction schedule has been achieved or all problems areas in the proposed schedule have been identified to the user.

During 1986, the infrastructure for this system has been created. This infrastructure consists of the project database, integration of the database with the computer graphics systems, methods for three-dimensional design, procedures for generation of three-dimensional construction sequence models, and software to determine construction material quantities from the three-dimensional models. Work is now under way on the development of the rule base and integration with the Stone & Webster project management system.

It is believed that this system, when complete, will provide construction planners with a better tool for creating the construction schedule from the engineering design, visualizing the construction sequence using computer graphics, and evaluating the constructibility of the plan using the expert system.

## **Expert Systems for Construction Scheduling - Research at Carnegie Mellon University**

by Chris Hendrickson and Daniel Rehak  
Department of Civil Engineering  
Carnegie Mellon University  
Pittsburgh, PA 15213

### **1. Introduction**

Both knowledge based expert systems and scheduling have been subjects of considerable research at Carnegie Mellon University. However, work directed at *construction project scheduling* is fairly recent. This short report will focus on the research contributing to the CONSTRUCTION PLANEX system. This system has demonstrated the feasibility of expert systems for construction planning and scheduling. It also provides a general architecture that can be adopted for different planning applications featuring the use of specific domain knowledge and conventional scheduling operations.

CONSTRUCTION PLANEX is a knowledge based expert system intended to synthesize activity networks, to recommend appropriate technologies, to estimate required resources (including activity durations), and to develop a project schedule. The knowledge in the current system pertains to excavation, foundations and structural erection for office building construction. The system is being implemented on a Texas Instrument EXPLORER™ in the KNOWLEDGE CRAFT™ environment. The prototype version of CONSTRUCTION PLANEX will be available in three versions: (1) a stand-alone aid for office building construction planning, (2) a component of a vertically integrated building design environment (including space planning, structural design and other considerations), and (3) a generic aid for project planning.

Contrasts are worth noting between CONSTRUCTION PLANEX and other planning models in artificial intelligence such as NOAH, NONLIN, DEVISER, and CALLISTO [2]. While these artificial intelligence based planning systems offer some extremely useful conceptual tools such as the general system of hierarchical activity representation in CALLISTO, each has significant limitations for construction planning. First, these systems generally incorporate only a relatively small number of well defined, repetitive tasks. In contrast, construction requires numerous distinct tasks for completion. Second, construction planning involves the selection of appropriate resources to apply, in contrast to blockworld or job shop scheduling problems in which resources are given. Third, construction has numerous important planning concerns with respect to time constraints, cost, equipment availability, environmental conditions, and spatial restrictions which are not considered by many existing planning systems. Fourth, the large size of construction planning problems suggests that efficient, algorithmic scheduling tools may be desirable rather than relying entirely on heuristic allocations. Fifth, construction planning is highly knowledge intensive, so explicit use of expert knowledge is required in the planning process. These observations motivated the design of the CONSTRUCTION PLANEX system to emphasize the use of both expert knowledge and algorithmic scheduling procedures.

While CONSTRUCTION PLANEX is intended for construction project planning and scheduling, we should emphasize that research in related areas is continuing at Carnegie Mellon and has influenced our

ideas on CONSTRUCTION PLANEX. Some related developments include:

- Refinement in appropriate software environments:  
The FRAMEKIT and RULEKIT utilities were used for the initial prototype of CONSTRUCTION PLANEX. This environment was abandoned to take advantage of the user interface facilities of KNOWLEDGECRAFT.
- Interaction with engineering databases:  
KADBASE demonstrated the capability of multiple expert systems accessing a distributed network of database management systems.
- Printing production:  
Expert Technologies, Inc., is developing a prototype system for printers, including estimating of job costs, selection of production plans and details, and print shop scheduling and management.

## 2. Architecture of CONSTRUCTION PLANEX

Similar to other knowledge-based expert systems, CONSTRUCTION PLANEX has familiar general components [4]: (1) a user interface, (2) a context, (3) a system control module, and (4) a knowledge base. Within these various components, specialized data structures and operators exist.

In the *Context*, information about the current plan is summarized in two hierarchies. The *design hierarchy* represents the various facility components to be constructed. The lowest level of the hierarchy represents work activities associated with individual design elements, and upper levels are aggregations of design elements grouped into structural components and systems. Much of the design hierarchy is input to CONSTRUCTION PLANEX, with the exception of quantities of materials required and element activities. A standard coding system of design elements is assumed [1]: CONSTRUCTION PLANEX will only plan activities for recognized design elements. The *activity hierarchy* also includes element activities at the lowest level, but upper levels represent functional aggregations of lower levels. Associated with nodes in the two hierarchies are PLANEX results such as technology choices, activity durations or material requirements.

*Knowledge Sources* comprise the bulk of information in the knowledge base. Knowledge sources include rules for (1) quantity-take-off from design elements, (2) element activity creation from design elements, (3) technology choice at different levels of the activity hierarchy, (4) duration estimation for element activities, (5) cost estimation, and (6) precedence setting. Thus, for each possible design element, numerous knowledge sources will exist. An early prototype of a knowledge source was the MASON expert system for estimation of the duration of masonry construction [3]; this estimation structure was formalized in the CONSTRUCTION PLANEX knowledge source model. Each knowledge source is a decision table or a network of decision tables intended to fill in the value of a slot in the system context. A special *Knowledge Acquisition Module* [5] was created to permit development of knowledge sources in a spreadsheet-like environment before translation into schema representations.

*Operators* are used to control the system's actions and to evaluate knowledge sources. A single knowledge source evaluator operator can be used for the various knowledge sources such as quantity-take-off, activity creation, technology choice, duration estimation, etc. Control operators are responsible for scheduling different planning activities in the absence of user direction.

*Scheduling* is achieved by an interactive, algorithmic operator in the system. Multiple precedence types, activity windows, and resource constrained scheduling are supported. Resource allocation is

performed heuristically. An interactive scheduling mode is available with screen displays of GANTT charts and traces of resource use over time. In this mode, the scheduled start time of particular activities can be specified.

### 3. System Status and Research Issues

The original system prototype demonstrated the feasibility of generating and scheduling construction plans using an expert system. The second, improved system prototype is now being coded. It will contain knowledge sources for planning excavation, foundation work and structural assembly of office buildings. The control operators are being improved to permit more efficient revision of plans. Comparisons with actual construction cases are planned during the next six months.

Some open research questions include the following:

1. How might the planning information generated by PLANEX be used during project control and monitoring?
2. What is the best architecture for control operators during revisions of plans?
3. What is the proper role for algorithmic resource allocation and scheduling versus local heuristics of resource choice and assignments?
4. Can a generic planning environment be developed to which domain specific knowledge sources are added?
5. What would be the field experience with a prototype system such as CONSTRUCTION PLANEX?
6. What is the appropriate expert system technology and system requirements to use to build a production system?

### 4. References

- [1] Baracco-Miller, E.  
Planning for Construction.  
Master's thesis, Department of Civil Engineering, Carnegie Mellon University, 1987.
- [2] Hendrickson, C.T., Zozaya-Gorostiza C., Rehak, D., Baracco-Miller, E. and P. Lim.  
*An Expert System Architecture for Construction Planning.*  
Technical Report EDRC-12-07-87, Engineering Design Research Center, Carnegie Mellon University, Pittsburgh, PA., 1986.
- [3] Hendrickson, C.T., Martinelli, D. and D. Rehak.  
Hierarchical Rule-Based Activity Duration Estimation.  
*ASCE Journal of Construction and Engineering Management*, June, 1987.
- [4] Zozaya-Gorostiza, C.A.  
*CONSTRUCTION PLANEX: Revision of the Architecture of the System.*  
Technical Report, Department of Civil Engineering, Carnegie Mellon University, Pittsburgh, PA., 1986.
- [5] Zozaya-Gorostiza, C.A.  
*CONSTRUCTION PLANEX: Knowledge Source Acquisition Module.*  
Technical Report, Department of Civil Engineering, Carnegie Mellon University, Pittsburgh, PA., 1987.

SUMMARY for the U.S. Army Construction Engineering Research Laboratory Workshop on Expert Systems for Construction Scheduling, March 23-24, 1987 in Champaign, Illinois

AUTOMATED PLANNING TOOL  
A Testbed for Knowledge-Based Project Management

Glenn M. Yoshimoto  
Knowledge-based Applications, Group Leader  
Lockheed Artificial Intelligence Center  
2710 Sand Hill Road, Menlo Park, California 94025  
Organization 90-06, Building 259

WORK ALREADY ACCOMPLISHED

Our knowledge-based planning and scheduling work has been in two areas:

(1) development of a planning/scheduling prototype for a knowledge-based Space Station Coordinator (an architecture for planning, execution monitoring and control, and anomaly handling), and (2) development of a knowledge-based project management system for software system development.

The second project was a joint effort between Lockheed's Software Technology Center (STC) in Austin Texas and the Lockheed Research & Development Division in Palo Alto California. This capability is to be embedded in a future-generation software development environment. Both prototypes contained a Critical Path Method (CPM) scheduling kernel. This talk primarily covers our Automated Planning Tool (APT) work of the second project but examples will be drawn from both projects.

Currently, APT capabilities represent a subset of those available in conventional scheduling tools. Our approach has been to start with the implementation of conventional CPM techniques within a knowledge-based environment to provide a testbed for the evaluation of knowledge-based project management representation and inference control schemes. This approach acknowledges that 2 decades of development and practice in project management have produced a standard and useful set of techniques, including representations and procedures. Our approach and progress includes:

- Adoption of CPM network representations and methods
- Extension of basic CPM precedence relations for delays \*
- Hierarchical activity and resource representation in schemata (semantic networks) \*
- CPM scheduling via production rules
- Resource requirements leveling based on the generation and testing of alternatives
- User interface display of multiple activity representation formats (network, Gantt, and tabular displays)
- Classification of activities and development of typical activity templates \*

Our effort in knowledge representation (noted by the asterisks) was to develop hierarchical structures that organize relational knowledge about activities and the resources they require. Human experts use associative memory networks (which encode complex relationships between memory objects) and have spreading-activation of memory traces for recall of relevant facts. It is difficult to represent such knowledge in relational databases and other conventional software. To start the encoding process, we have developed a layered, relational knowledge structure for the organization of project management knowledge. In addition, we have found by studying simple examples of project management problem-solving that large amounts of knowledge will be required to provide a generally useful automated project management system.

Model-based reasoning appears appropriate for robust automated activity network generation, measurement, and diagnosis. Physical and social (management) process models for specialized project types can be developed to support intelligent project management; these we have started to conceptualize. At Lockheed, we have tended to study "hard" knowledge engineering problems since our entry into AI. By "hard", I mean the scale of systems that automate operations planning of large systems such as the Space Station and large military C3I embedded software systems. The scale of construction project management systems is the same order-of-magnitude and encompasses the knowledge areas of construction design, federal regulation, environmental constraints and impacts, geology, cultural anthropology, foreign policy, construction materials (kinds and availability), subcontractors and capabilities, and international logistics.

#### ONGOING RESEARCH EFFORTS

We have many research projects that are scoped to provide technology for solving our "hard" problems. These results will also apply to knowledge-based management of complex projects:

- Distributed concurrent architectures (blackboard architectures distributed across a network of workstations with emphasis on open system philosophy and flexible and evolvable knowledge-based control and system modularization); these results will permit the controlled distribution of complex, concurrent reasoning processes over many processing elements.
- Model-based reasoning (structural and behavioral modeling of spacecraft, and C3I threats and assets); these results will provide techniques and tools for encoding expert models.
- Integrated knowledge-based workstations and software development environments (C3I and Express); these results will provide architectural insights, techniques, and tools for the implementation of large scale knowledge-based systems.

#### UNEXPLORED AREAS FOR RESEARCH

Project management techniques have been found to be invaluable in developing the initial logic of an activity and serve as a means for

a collective process of thinking about the project objectives and the strategies and activities that implement them. Many potential benefits of knowledge-based systems technology exist, but these systems will be difficult and costly to develop because of the large amount of knowledge involved. The basic issues are:

- We lack effective means to index computerized data with complex relationships in a way that is useful to a large population of users (for example, cost/schedule history, technology, design, and programmatic information and status).
- A large effort is required to maintain comprehensive networks and other project control information after their initial development. (The logic and justifications that initially went into them are difficult to represent and preserve for future reference.) It is crucial to achieve this.
- Interactive tools that are useful for schedule development are not well integrated into the tools and systems that are being used to control activities; they are also not generally used by managers.
- The impact of potential risks in activities is difficult to account for in project planning. Effective techniques and tools for risk assessment and preservation of decision justifications have not been established.

These issues seem to result from social rather than technological failures (development-participants typically resist project management systems and their direct use). Knowledge-based systems technology may provide solutions by providing much richer stores of knowledge that are available upon demand in forms that are useful to a large variety of users. To succeed in knowledge-based initiatives of this scale, it seems clear that the scale of creative thinking must be at the level of integrated development environments. The knowledge encoded must be an integrated system of knowledge.

Research in integrated (special-purpose) knowledge-based development environments has started in some areas. At Lockheed, our STC software productivity initiative is developing a far-term, knowledge-based environment for the systematic, end-to-end development of software systems. The environment will be complete with technological knowledge and information, design specification languages, automated compilers for these languages, design analysis and simulation tools, verification and validation systems, documentation generation utilities, configuration management utilities, and project management and controls. The requirements for this environment are specified from a multi-perspective user view. The total collective knowledge about a field requires development of a languages in which all project participants can express their requirements, monitor results and interactions, and communicate. This language would be executable on a system of workstations that comprise the development environment. The system will support end-to-end development with respect to project phase (bid and proposal conceptualization through operations) and will provide diverse user support in the vertical direction (through management levels) and the horizontal direction (through disciplines).

The aims of this research are to develop languages and tools in which problems can be represented and solved in ways that correspond directly and naturally to our own conceptualizations.

# **Construction Knowledge Systems: Status of Work at The University of Texas**

Prepared for the USA-CERL  
Workshop on Expert Systems for  
Construction Scheduling, March 23-24, 1987

*by*

*David B. Ashley*

*The University of Texas at Austin*

*Austin, TX 78712*

## **Introduction**

Three summaries of ongoing expert system development work at The University of Texas are provided. All three are Civil Engineering oriented. The first two are direct outgrowths of the author's research on project success factors and problem databases. The third is a wood engineering system designed to assist designers in modifying design parameters for environmental conditions. All share a focus on the content of the knowledge bases.

## **A Knowledge Base for Repeating Construction Project Successes**

One of the recurring themes in construction industry research is how to improve efficiency and cost effectiveness. Repeating construction project successes by recognizing their determining factors is the goal of this proposed expert system. Through a comparison between data of average and outstanding projects it is possible to identify a variety of factors that differ significantly between the two classes of outcomes. Additional analysis also demonstrates how these factors affect budget and schedule performance.

A database containing previous project data and research analysis results is used as part of the system. As additional projects are completed and added to the database the analysis results are automatically updated. The proposed expert system uses the developed knowledge base and other relevant data to seek opportunities for

improvement for a proposed new project. The system estimates the likelihood of achieving an outstanding outcome for each considered strategy. Using resource constraints and project objectives as additional inputs, the expert system guides the user toward a preferred planning and execution strategy.

This system is being developed by David B. Ashley, Edward Jaselskis, and Prapat Tantiprabha. Initial efforts are on structuring the correlation, logit regression and discriminant factor analyses used to update results. These statistical routines are linked via C language hooks to an M.1<sup>TM</sup> expert system shell. The expert system is thus envisioned to have a modest amount of self-learning.

### **IRIS: An Intelligent Construction Risk Identification System**

Risks and uncertainties can arise in any phase of a construction project. Effective management of these risks is essential for a successful project. The goal of this expert system is to help construction managers identify, analyze and control the possible problems they might face in a construction project. **IRIS** is an expert system designed to help construction professionals with the first important task of risk identification.

The architecture of **IRIS** consists of an extensive database of construction problem statements collected primarily from interviewing experienced construction personnel and other experts, a deductive inferencing mechanism for reasoning and a graphical routine for displaying the risk relationships. The C programming language is used to integrate the deductive inferencing, database management, and graphical representation functions. The functions within the system are built around M.1<sup>TM</sup>, **RBase System V<sup>TM</sup>** and **Multihalo** software packages. Information available in the database includes issues with potential cost impact and schedule delay, cause-effect relationships of these issues, certainty factors for these relationships, effective and ineffective management actions, and impact of these actions.

A rule-based knowledge representation is employed to handle the reasoning and work together with the query search of the database management system. The system decides which data files should be included in a search for applicable problem statements. Basic influence diagrams for each identified problem are drawn automatically. The user can interactively add or delete risk factors on this influence

diagram. These modifications can be retained by the system; thus there is modest learning by IRIS. Once an influence diagram is developed and verified as an accurate model of the problem under investigation, it can be automatically carried forward to a sensitivity analysis. Using Bayesian inference techniques this influence diagram can generate a monitoring/control scheme to allow a manager to track identified risk factors. Another implication of this approach is the automatic generation of a diagnostic expert system to analyze cost or schedule overruns.

IRIS is being developed by David B. Ashley and Y-H Perng.

#### **WOOD: Knowledge Base for End-Use Design Stresses in Wood**

Proportioning of wood structural members is currently based on a working stress design procedure. Allowable stresses must be modified by end-use factors if the temperature or moisture environment while in service are different from a prescribed set of conditions. In practice many engineers are unaware of the specific conditions which would necessitate the use of these factors, or they are uncertain of when or how to interpret the code requirements about the use of these factors. **WOOD** is a prototype expert system designed to demonstrate how the uncertainties associated with the environmental conditions may be incorporated into the design process so that the engineer will have the proper set of allowable stresses to begin proportioning the wood members.

**WOOD** queries the user about the anticipated design environment, inquires about how certain the user is of this information and then recommends factors by which the allowable stresses should be modified. Recommendations are in the form of decimal multipliers of the published allowable stresses. Allowable design stresses included are:

1.  $F_b$  = extreme fiber in bending;
2.  $F_t$  = tension parallel to grain;
3.  $F_c$  = compression parallel to grain;
4.  $F_c$  = compression perpendicular to grain;
5.  $F_v$  = horizontal shear; and
6.  $E$  = modulus of elasticity parallel to grain.

A different modification factor applies to each allowable stress because the various associated strength values are affected to different degrees. The expert system does

not follow code guidelines strictly. Rather, it uses established data on equilibrium moisture contents -- predicted from relative humidity and temperature data -- as well as more recently published research results on the mechanical properties of wood as a function of both temperature and moisture content, to generate end-use factors. It is intended as a demonstration of how the design engineer's knowledge of the end-use of a structure may be coupled with the researcher's knowledge about the behavior of wood in adverse environments to result in a properly designed wood structure.

This system is the joint work of Prapat Tantiprabha, Dan L. Wheat and David B. Ashley. It is currently implemented in a M.I.<sup>TM</sup> environment. Future work is directed toward: 1) expanding the wood research knowledge in the system, 2) developing probabilistic interpretations of the wood research results and incorporating them in the system reasoning, and 3) system validation.

#### Comments

It is too early in the development of these systems to predict with absolute confidence their successes. The first described system is perhaps the most ambitious. It presupposes that there are common, underlying factors among outstanding projects that distinguish them from the average. It uses derived predictive models to provide a reasoning core for the knowledge system. Validation of this system will be a complex task. **WOOD**, on the other hand, is more closely linked with engineering practice. It is easy to see how this system might interface with design. Validation should be a straight-forward task.

As mentioned in the introduction, all three development efforts focus on knowledge content. The project success and **WOOD** systems are envisioned as vehicles for better research dissemination. Both the success and **IRIS** systems concentrate on continual expansion of the databases and how best to incorporate new knowledge.

## Research at M.I.T. on Application of Knowledge Based Systems to the Project Control Process

Robert D. Logcher\*

### INTRODUCTION

Attached is a paper entitled "Adding Knowledge Based Systems Technology to Project Control System," prepared by myself for the upcoming ASCE Specialty Conference promoted by Bill Ibbs. In this paper I make the point that our current project control systems deal with projects and construction almost exclusively in a genetic sense. That means that they can accept and regurgitate massive amounts of data without knowing much about their meaning. The care and feeding of these systems, however, is highly knowledge intensive, leading to the need to manually introduce such knowledge into the application of these systems.

This problem is most clearly seen in the network scheduling problems. Here we have some commonly used and very simple algorithms which will calculate and recalculate schedule dates and schedule status if only the user will go through the following steps:

1. Identify all individual tasks required
2. Design task execution
3. Determine resource requirements including time
4. Determine task sequencing
5. Determine resource schedules, calendar, etc.
6. Determine constraints, timing and resource
7. Study project status data and develop activity status
8. Determine the implications of status on the validity of the plan
9. Determine why (and because of whom) status differs from plan

\*Professor of Civil Engineering, M.I.T., Cambridge, MA 02139

10. Determine how differences effect rest of tasks, resources, etc.
11. Determine how acceptability of these effects
12. Determine how to modify plan to make it acceptable
13. Determine how to modify plan to make it less vulnerable to the causes of differences

Not only are these processes knowledge intensive, dealing with knowledge of the domain of construction technology, they are also data intensive, dealing with as broad a set of project data as are available at any time. Data can no longer be thought of as scheduling data and financial data, but must be integrated in the broadest sense. This points out the reason for the title of this document, dealing not only with scheduling, but in an integrated fashion with all of project control.

On the other hand, scheduling is a simple, well-understood subset of the project control process. We can deal with it as a learning tool, to test out concepts and structures for our broader systems. But in doing so, we must be careful not to take too a narrow view of the problem. While our inferencing might deal only with schedule implications, we must structure our systems to use the more complete concepts of applicable knowledge and broad project data.

We have been working at M.I.T. for almost three years on the development of knowledge based systems for project control. We have been working on scheduling problems per se for almost two years. Early work was on conceptual systems and later implementations carried out during the past 1-1/2 years using KEE on a TI Explorer and home-built shells on VAX's and PC's.

#### EARLIER EFFORTS

The first effort directly related to this research topic was the conceptual design of a knowledge base for the identification of the causes of variance from plans. This topic was tackled in 1984, early in our efforts because we felt its solution was essential to solving the time/cost creep problem and to provide better forecasting. This was a paper scenario of the operation of an object oriented knowledge structure. Niwa and his group at Hitachi had also tackled this problem and given up due to inefficiency in the inferencing process. The scheme proposed used preidentified

risk factors, each with a rule base for analyzing project and work package sensitivities. Individual work packages would then be ranked, tying them to likely factors. When progress differed from expectations, the inference process would use the sensitivities to hypothesize causes. Similarities among sensitivities could then be used to support or refute the hypothesis. A dialog with the user was felt necessary to guide the process for efficiency, utilizing the user's judgment, and dealing with new and low sensitivity risk factors.

At this point in time, we have not attempted to implement these concepts. We are still developing some more basic tools and components. This work does point out that the quantity of data and knowledge needed for solving this type of problem is very large, and thus requires great care in knowledge structuring. The concepts of object oriented programming are really required for this type of problem.

The paper mentioned in the attached paper, [Navinchandra and Logcher 1986] supports this point. This work, done in the Spring of 1985 on a VAX 11/750 in Franz Lisp using a home-built shell called IMST, dealt with the analysis of a simple job cost report. IMST, an expansion of OPS5, dealt with this database problem using a partitioned set of rules. Yet, even with a modest amount of data and performance assessment objectives, performance of the system was a problem. Inferencing time was excessive.

In late 1985 and early 1986, Mauricio Arias-Toro and Juan Carlos Aldana tackled the problem of schedule generation. Both dealt with planning schedules, one for a department of public work, one for the US Army military construction process. In both cases, large amounts of regulation and procedural knowledge was available. Major problems existed in variances in the characteristics of the project over its life and the need for earlier prediction of likely delays and cancellations. Knowledge and data analysis for such predictions needed to be included. This project was aimed at both schedule generation and updating (both progress recognition and schedule structure changes). But, as Master's theses, we didn't get as far as planned. An implementation was completed in KEE of a knowledge structure which generated and floated schedules based on project characteristics. Subnetworks were connected together using rule classes associated with each subnet. So far this structure is much too simplistic. The simple rule classes needed to know the base network structure into which they were inserting their subnet.

This work used a method or algorithm for floating the network. This is efficient, but does not provide the flexibility, particularly in updating, that I was looking for. This has now been corrected in our current work. This work also started to develop a rule base for interpreting progress on a project from non-schedule data and automatically updating the schedule.

#### CURRENT SCHEDULING RESEARCH

Five efforts are currently underway, three of which will be completed by this summer. The first, mentioned in the attached paper, is an implementation of the CPM algorithm in KEE using pure message passing. In doing this, any object is independent of any other except in terms of the information it has about others. Messages are received by an object, which knows how to process them using only its own knowledge. It then sends messages to others impacted by its inferencing. When first implemented in KEE, we got "stack overflows". This was solved by implementing our own blackboard for message control. The process is implemented for schedule changes and updating as well as initial scheduling, with schedule changes propagating only as far as they change information. The process runs as fast as an algorithmic approach and has the advantage to considering of numerous very small methods inherited from generic objects and which could easily be modified and specialized. (Example in paper on duration calculation for time of year.)

Given this tool, we are now implementing a knowledge structure with daemons which run around an object representation of a database looking for data changes and inferencing about the schedule implications of these changes. We will be including trend analysis in the knowledge processing within these daemons. With the previous work, the schedule changes are automatically propagated.

We are also implementing resource constrained scheduling within this KEE knowledge environment. While at present the approach does not differ from common heuristic algorithms, we are using branch and bound generate and test methods on partial solutions as part of the heuristics. Efficiency problems are not known yet.

Next I want to mention GHOST, a blackboard architecture for determining schedule precedences given activities and their environment. This system started with all identified activities in parallel and then used critics such as physical conditions and construction technology to introduce additional precedences. Redundant precedences could then be

removed by another critic. Subnet introduction is used for hierarchical refinement, but we still need to work on criteria for the use of refinement and better tie-ins for the subnets so that the schedule duration is reduced by the refinement (i.e., overlapping introduced).

Lastly, in conjunction with our work on building construction robots, we are developing a robot planning system which will schedule the robot motions and coordinate their activities with interfacing trades. Our first suite of robots is for gypsum wallboard partitions, track, studs, and board, so this planning involves most of the finishing trades. This effort is just getting underway.

#### OVERALL OBJECTIVES

I am also including a position paper I wrote last Fall on the thrust of M.I.T.'s CE knowledge based systems efforts. It might help clarify the role of construction management in a larger design/construction process.

Prepared for the Proceedings of the ASCE Specialty Conference  
on Integrated Project Controls for Design and Construction  
Lincolnshire, IL June 8-9, 1987

Adding Knowledge Based Systems  
Technology to Project Control Systems

Robert D. Logcher, F. ASCE (1)

Introduction

Computers have been actively used in civil engineering design and construction for over 30 years. They have promoted improvement in productivity and effectiveness of the construction product. It is easy to point to areas of success, such as financial project control systems, network scheduling, accurate and detailed analysis procedures, automated component design procedures, etc. While these applications have changed our "way of doing business", they have also created a new set of problems, or shall we say, opportunities. When we have solved these problems, we will have created a new generation of project control systems which will provide far more effective tools for our industry. This paper will present a series of problems and describe how emerging new technologies can be used for their solutions.

The principal new technologies with which this paper deals all come from the disciplines of Artificial Intelligence. They include Knowledge Based (Expert) Systems (KBS), knowledge representation, object oriented programming, and natural language interpretation. These tools, not currently utilized in our project control systems, provide opportunities for drastically altering the character of these systems.

The computer has had some very deleterious impacts on our profession. In the same way as numerically controlled machines have de-skilled the machine tool industry, computers are de-skilling engineering design and management jobs. Already we see little need for a deep understanding of structural analysis and component design in the engineering office. These tasks have been automated. All the more reason why we need the next generation of tools, tools which can retain and exercise the knowledge no longer required of the designer. We must now deal with the analogies in design and construction management.

(1) Professor of Civil Engineering, Massachusetts Institute of Technology, Cambridge, MA 02139.

### Early Capture of Information

This author has a bone to pick with much of the software industry which produces control systems for our industry. Most of these systems are generic and have evolved from financial accounting systems. In attempting to remain generic and deal only minimally with non-financial information, these systems have constrained their utility to historical record-keeping with little in the way of early warning or forecasting of problems. This leads to slow and late recognition of problems.

The issue revolves around both the early capture of data in a database so that it can be used for forecasting purposes and the breadth of scope of that data. It is well exemplified by using commitments from purchase orders and subcontracts to forecast cost at completion, variance from budget, and remaining exposure in job cost reports. Many systems don't use such commitment data. Also, payment requisitions forms, when sent to the field for data capture and entered into the computer, can provide immediate progress data while producing the invoice for the field. Input of a rough estimate with the identification of the need for a change order tracks both this need for the change order and its financial impacts.

The solution to this problem is embodied in more integrated systems, systems coordinated with better information flow procedures within companies, and a broader view of information content in the project control process. It is this latter solution component, a broader view of information, that is required if we are to utilize the new technologies mentioned above. For example, the mention of the need for a change order on a job, having its inception suggested by a particular subcontractor, may suggest a host of financial and management problems to the experienced project manager. Our future systems will attempt to capture this experience and mirror the reasoning of the project manager.

### New Technologies

In this section, the technologies are mentioned and very briefly explained. They have not been invented by civil engineers, so we will not let them dominate this paper. Rather, our contributions lie in their application to our problems. Herein lies our challenge. Yet, it is worth mentioning that the character of our problems often challenges such technologies. Our problems tend to be larger, involving more disciplines and interactions, then the financial, mechanical engineering, and even VLSI

industries. We tend to have larger databases, larger KBS's, more detailed CAD drawings and design data, and so on. As such, we will constantly be pushing these technologies.

The first technology in knowledge based systems. Fenves [Fenves, 1986] defines such systems by a separation of a knowledge base from a control or inferencing scheme which uses the knowledge, and an ability of a system to explain how the knowledge was used to reach conclusions. In these systems, domain dependent knowledge is usually stored in rules which should be understandable to the domain expert. (Note: the author uses the word "expert" guardedly, since real expert behavior is hard to corroborate, leading to more modest goals for most systems, and many systems now under development use multiple knowledge sources, including all of their users. In such cases, maybe we should call them apprentice systems.) With the separation of the knowledge base, these systems are easily changed to allow incremental growth.

Problem solving in these systems use a combination of forward and backward chaining through rules. In forward chaining, the rule base is checked using problem data for rules for which the premise is true. The action or conclusion parts of the true rules provide new data which might cause other rules to become true. When no more rules fire, the process ends and the data contains the solution. With backward chaining, a hypothesis of the solution is first generated using forward chaining, and then attempts are made to verify the hypothesis by looking at rules which contain the hypothesis in their conclusion and seeing if their premise are true. If all data in their premise is not known, the process chains backward by attempting to determine such data in the same manner. If the backward chaining finds data to verify all premises in the chain, the hypothesis is true, and if not, false. Then another hypothesis must be generated and tested.

Pure rule based systems are only useful for small problems where all the knowledge can be represented in several hundred rules. The problem with rule based systems is that their efficiency degrades exponentially with knowledge base size. [As shown in Niwa, 1984] As the breadth of knowledge increases, the premise of the rules become longer and more complex in order to apply the rules to their appropriate subset of the problem domain. Checking for applicable rules also takes longer as the knowledge base size increases. Since we expect our systems to become integrated and very large, this form of system is will not work acceptibly. We must therefore look toward the concepts of object-oriented programming [Abelson and Sussman, 1985] and frame-based systems. A frame or object is analogous to

a record in a database, with numerous additional characteristics. One object may be a pattern for its instances, where the instance objects then inherit its slots, or knowledge holders, and slot characteristics. Slots may hold data, and slot characteristics might include a data verification procedure, of which acceptable bounds is the simplest example. Slots may also be programs or procedures or knowledge such as rules or rule bases. Objects may be related in more complex manners analogous to the set relationships in network databases. The concept of inheritance may then be associated with any relationships so that objects can take on properties of several objects to which they are related. An activity can inherit an algorithm for calculating its early start and finish dates given those of its predecessors from a generic CPM activity. It could also inherit knowledge about how to figure its duration from knowledge about its type of work. Finally, slot characteristics may include procedure or active rule bases. Then, when a data value in a slot is changed, this automatically triggers actions.

The concept of message passing is central to object oriented programming. One object, operating independently, can reach some conclusion and then inform other objects to which it is related of this conclusion. Messages sent to an object cause it to store data and initiate procedures which check the impact of the message on itself. This may result in changes within itself, changes which may in turn generate messages to other objects. A blackboard, or message and data coordinator, is often used to control reasoning processes. The section after the next in this paper provides a detailed example of the use of these concepts.

The application of such features is quickly apparent. Rule bases can now be disaggregated, leading to efficiency and disaggregate collection of knowledge. Procedural knowledge can be conveniently mixed with other forms. If we think of a building design stored in this way, when a specified pump is unavailable during construction, the field personnel provide a cable of available pumps, the pump object redesigns itself and then signals its interfacing technologies, electrical and structural, of the interface changes, which are then checked and, if necessary, changed or their designers signalled of a required change. Change then propagates, providing change management.

The frame based structure provides an effective tool for planning paradigms. Scenarios of conditions, actions, and their effectiveness could be stored as an historical knowledge base. When a planning problem is recognized and immediate knowledge not sufficient for a solution, the knowledge base is searched for similar conditions and

alternative actions thereby generated for further screening. This structure may be more efficient than the generate and prune algorithms using a single fixed knowledge base.

Lastly, we can apply these techniques to create intelligent database systems. Such systems have daemons which the system scheduler starts on a periodic basis. They traverse the database, looking for changes in data since their last pass. When finding changes, a daemon would start an inferencing process to check for impacts of the change and look for patterns of changes and causes. This might then initiate forecasting and planning processes, thus leading to much more dynamic project control systems.

Natural language translation is obviously a technology closely linked to all applications. Its use is for more natural, user-friendly communication with these systems, to provide knowledge capture, data input, and processing control. Its application together with KBS's is most interesting. While numerous natural language query systems are available, often closely related to database managers, such as CLOUT with R:Base or INTELLECT with FOCUS, their domain is realistically limited to direct retrieval and minor manipulation of data from one or more files. Dictionary words are associated with individual fields or groups of fields in a file or very simple direct operations on the fields. New systems, such as Expert-MCA being developed at M.I.T., use deeper user-defined knowledge of the meaning of data to answer more complex questions. [Logcher, 1986] Such knowledge defines pattern searches against the database.

#### Database Data Analysis

When we look at our project control process, we are struck with the realization that the computer tools we use are so predominately generic, having almost no knowledge of project, company, or construction technology, that we must introduce very large amounts of manual processing to gain information out of our systems. Our CPM schedules are generated for every project from scratch, or maybe from a previous similar project. Our job cost budget may be generated from our estimate, which was similarly generated. We are starting to see some estimating systems [QuickEst, 1985] which use cost modeling techniques, where a component model does incorporate design and construction knowledge. This section and those that follow try to provide examples of how such knowledge can change our project control process. The next section shows how the technology of object oriented programming can help in the implementation.

With our database technology, we have been able to collect, process and report very large amounts of project data very easily. As a result, we have made decisions to disaggregate our project down to the smallest responsible party. We can then report accomplishments and compare against expectations for all parts of our project. We then overwhelm users with data. (Note: the author does not say "information".) Exception reporting might keep down the volume of results, but does little to help us understand the causes of our problems. This is still a hard, knowledge intensive process, hampered by having a narrow window into our project in which updating takes place (small percentage of accounts).

IPMS [Navinchandra and Logcher, 1986] attacked this problem. Taking a typical job cost report with responsibility for an account shared between estimator, superintendent, and foreman, it showed how a rule base could be used to search for patterns among accounts to evaluate performance. Data needs to be conditioned to eliminate estimator bias before field personnel can be evaluated. Determining whether the poor performance of one foreman was the fault of the foreman was found by looking at the performance of other foremen working under the same superintendent. A pattern of poor performance indicated that the superintendent was most likely at fault. Figure 1 shows a typical rule in this system, while Figure 2 shows some typical results. An explanation capability is included.

#### Example Application of Object Oriented Programming

It is worth illustrating the character of this technology through a project control example. While the example, calculation of dates in a CPM schedule, could easily be solved with a simple algorithm dealing with all the network data together, the application of the concept of disaggregated knowledge become clearer. This application has been implemented by the author and his students in KEE [IntelliCorp, 1986] on an AI workstation.

In its simplest form, the system contains generic objects called PROJECT, ACTIVITY, and RELATION. Each of these contain slots for the data typically stored in algorithmic programs. In addition, these objects contain some very simple procedures, each consisting of small steps from the algorithmic solution process. When we solve the scheduling process with pure message passing [Abelson and Sussman, 1985] we will store some data in the objects which are duplicates of data in other objects, but we will show the value of this duplication.

```

Rule:      (for CaseIX)

IF      <the account has ACOST reported>
AND     <the AQTY is reported>
AND     <the REVQTY is reported>
AND     <the ETCCOM is not reported>

THEN    <it may be concluded the the account is
        a CaseIX account>

Rule:

IF      <an account has high actual costs>
AND     <the ESTIMATOR has a history of low estimates>
AND     <the SUPERVISOR has a good history>
AND     <the FOREMAN has at least a moderately good
        history>

THEN    <it may be concluded that the ESTIMATOR is at
        fault>

```

Figure 1. Typical IPMS Rules.

The foreman David Brown shows a tendency to overspend by 30 % while working for super Tom Fulton, who also has a tendency to overspend

Estimator Mark Wilson underestimates by 12.5%

Foreman David Brown overspent on ACCT 35, but did OK because estimate was too low

Figure 2. Typical IPMS Results.

The generic PROJECT contains a procedure called CREATE.PROJECT. Whenever this procedure is initiated (by sending it a message), it will prompt the user for project data, name, start date, finish constraint, etc., create a project instance object called PROJECT.name, and then allow the input of activities and relationships. The activities are instances of the generic activity and are related to the project instance. Data stored in the activities and relationships are durations, lead/lags, and network topology.

The algorithm is initiated by sending a message to the INITIAL.SCHEDULING procedure in the object PROJECT.name. The procedure is not stored in the object, but inherited from the generic PROJECT. This procedure sends messages to each activity telling each to initiate itself (set early and late dates to NIL, erase any dates from predecessors and successors). This message contains the project start and finish dates. For simplicity, we will assume that the project has a specified finish date. Activities receiving this message can handle it without needing information from other objects. If an activity has no predecessors, it can take the project start date and schedule itself. It then send messages to successor relationships informing them of its schedule. The relationships in turn send messages to their successor activities. The activity, receiving a message from a relationship, stores the message data and decides if all predecessors have reported. If so, the activity can be scheduled and the process propagates itself to the terminal activities. If an activity has no successors, it uses the project finish date in the same manner and proceeds with the backward pass at the same time as the forward pass. Float is calculated by whichever pass processes the activity last.

While this might seems like a complex process, it is simple and efficient. The procedures are small and inherited from the generic objects, not duplicated in each object. The real benefit comes when we realize that with this same structure, we can maintain a schedule during progress reporting. With only slight changes to the activity procedures, the impacts of progress reports can be propagated forward and backward only as far as they change schedule dates. We are using this with a knowledge base which analyzes project data (Note: not CPM data) such as charges to cost accounts or down time for a particular piece of equipment to send messages to activities about actual starts, finishes, changes in duration, etc. From these changes, recalculation is automatic.

To extend this concept further, we might consider the procedure for calculating the early dates for a activity. The procedure in the generic activity is simply to add the duration to the start date if a finish date is not determined from a start or finish to finish relationship. We could, however, substitute for this generic procedure a smarter procedure that understood that the productivity of the particular activity was weather sensitive and that the duration or finish calculation should be a function of the time of year of the activity start. This procedure could be inherited based on the type of activity.

The point of this example is that this technology allows us to insert easily into the control systems knowledge and relationships about our data and use it directly in forecasting and problem detection.

### CPM Scheduling

It is easy to see that existing CPM systems do not solve our real project management problems. Problems abound, from the tremendous amount of knowledge that must be applied to generation, updating, and interpretation of schedules to the well-known time/cost creep problem. The development of KBS's using object oriented programming techniques are geared solving these problems. These techniques have been applied recently in various research efforts. [Arias-Toro, 1986, Levitt, 1985, Hendrickson, 1986] The problem can be decomposed into several parts, each needing different inferencing techniques. These parts are:

1. Schedule generation - task identification, task design, and sequencing
2. Progress reporting - automatic schedule updating from database data
3. Analysis of variance - determining the cause for performance outside of expectations
4. Projection of variance onto remainder of schedule
5. Replanning to overcome impacts and mitigate causes of variance

Schedule generation is a planning process which can use many of the planning paradigms. Arias-Toro developed a knowledge structure which ties project characteristics to subnetworks with rule bases. A rule base is fired by the existence of a characteristic and is used to set subnet activity characteristics and connect the subnet into the project network. Physical, geometric, and construction technology constraints on the interactions between activities [Logcher, 1987] can also be used to generate precedences. These works also show how hierarchical detailing may be used to increase the detail in schedule design when needed to achieve schedule goals.

Currently, progress reporting involves manual input of activity starts, finishes, percent complete, and remaining durations. This data is abstracted from the broad

information available about the project. Using the concepts of intelligent databases and domain knowledge in the daemons, this process will be automated. The more difficult problem is determining the causes for variance. A preliminary knowledge structure for this analysis is shown in Figure 3. [Nay and Logcher, 1985 and 1986] Knowledge of potential causes of variance must be precoded, with rules embodying why an activity or work package might be sensitive to the cause. When a project is designed, each work package can be analyzed for its sensitivities. When variances are noted, these sensitivities represent first hypotheses for causes. Hypotheses are varified or refuted by checking other work packages with similar sensitivities as well as communicating with the project manager for outside influences and his ideas.

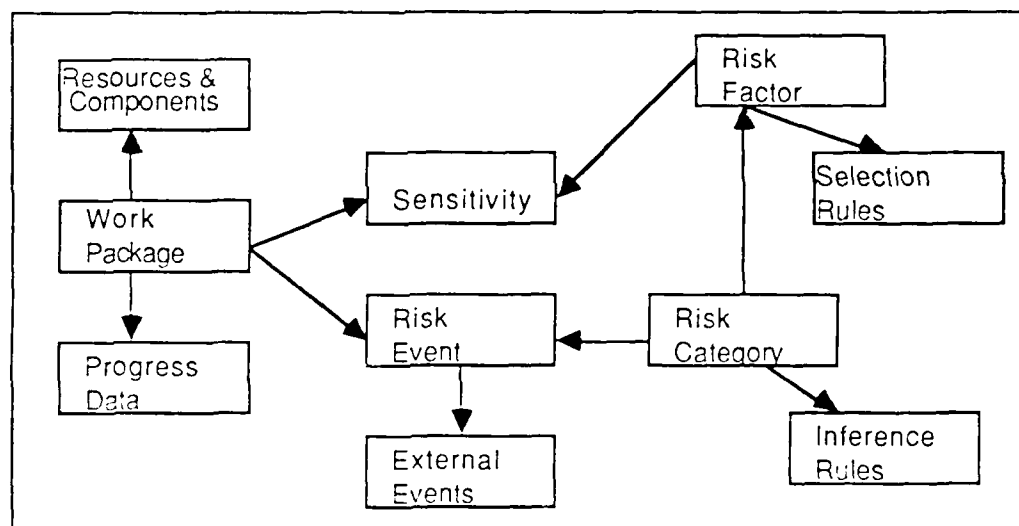


Figure 3. Knowledge Structure for Analysis of Variance.

### Conclusions

The applications mentioned here are only a start. Financial control, while not discussed specifically, has a direct analogy with schedule control. Throughout the

discussion, it is apparent that a separation is inappropriate and unnecessary. Our future systems will be far more integrated and deal in a common manner with all types of project, environment, organizational and other data, using knowledge to integrate and utilize the data.

#### References

Abelson and Sussman [1985] Abelson, Harold and Gerald J. Sussman, Structure and Interpretation of Computer Programs, M.I.T. Press, Cambridge, MA, 1985.

Arias-Toro [1986] Arias-Toro, Mauricio, Knowledge Based Planning Schedules for Public Works Projects, unpublished S.M. thesis, Department of Civil Engineering, M.I.T., Cambridge, MA, May 1986.

Fenves [1986] Fenves, S. J. "What is An Expert System," in Expert Systems in Civil Engineering, Proceedings of the ASCE Technical Council on Computer Practices, Seattle, WA, April 8-9, 1986, ASCE, New York, 1986.

Hendrickson [1986] Hendrickson, Chris, Carlos Zozaya-Gorostiza, Dan Rehak, Eduardo Baracco-Miller, and Peter Lim, An Expert System Architecture for Construction Planning, Engineering Design Research Center Report EDRC-12-07-87, Carnegie Mellon University, Pittsburg, PA, December 1986.

IntelliCorp [1986] KEE Software Development System User's Manual, KEE Version 3.0, IntelliCorp, Mountain View, CA, 1986.

Levitt [1985] Levitt, Raymond E. and John C. Kunz, "Using Knowledge of Construction and Project Management for Automated Schedule Updating," Project Management Journal, Vol. XVI, No. 5, pp. 57-76, December 1985.

Logcher [1986] Logcher, Robert D., Ayman M. Hindy, and Ming-Teh Wang, ESCUPAD: Initial Knowledge Based System for Processing Queries to CAPCES, Research Report R-86-10, Order No. CCRE-86-3, Department of Civil Engineering, M.I.T., Cambridge, MA, 1986.

Logcher [1987] Logcher, Robert D., D. Navinchandra, and Duvurru Sriram, "GHOST: A Project Schedule Generator," submitted for Expert Systems in Civil Engineering, J. of Technical Council on Computer Practices, ASCE.

Navinchandra and Logcher [1986] Navinchandra, D. and Robert D. Logcher, "Steps towards Automatic Interpretation of Project Data: A Knowledge Based Systems Approach," in Microcomputers in Engineering, B. A. Schrefler and R. W. Lewis, ed., Pineridge Press, Swansea, U.K., 1986.

Nay and Logcher [1985] Nay III, Leston B. and Robert D. Logcher, An Expert System Framework for Analyzing Construction Project Risks, Research Report R85-4, Order No. CCRE-85-2, Department of Civil Engineering, M.I.T., Cambridge, MA, 1985.

Nay and Logcher [1986] Nay III, Leston B. and Robert D. Logcher, "Proposed Operation of an Expert System for Analyzing Construction Project Risks," Proceedings of the Nineth Conference on Electronic Computation, Structural Division, ASCE, February 23-26, 1986, pp. 65-76.

Niwa [1984] Niwa, Kiyoshi, Koji Sasaki, and Hirokazu Ihara, "An Experimental Comparison of Knowledge Representation Schemes," in AI Magazine, American Association for Artificial Intelligence, Vol. 5, No. 2, pp 29-36.

QuickEst [1985] QuickEst Version 2.00 Estimating System, Program Description/Operations Manual, Constructive Computing, Kansas City, KS, 1985.

## Thrust for Research Program in Computation for the Center for Advanced Construction Technology

by Robert D. Logcher

### Overall Goals

The following research program is focused on the development and application of advanced computation tools which can be applied directly to improve the effectiveness and productivity of construction. Construction, as the downstream end of a larger development process, is strongly impacted by the characteristics and decisions made in the earlier steps in this process, from planning through detailed design. Computer tools, therefore, must deal with all parts of this process.

Major improvements in construction can be achieved using computation to promote:

1. Less error in design,
2. More detailed design,
3. Automation in construction,
4. Better construction planning,
5. Easier recognition of design and construction problems requiring decisions, and
6. Use of constructability criteria throughout design.

The research program deals with the nature of innovations in computation required to promote these goals.

### Background

Construction creates in general one-of-a-kind products which are unique configurations of widely used components. What components are included in the product is decided during an iterative design process which is heirarchical in nature, going from less detail about the characteristics of the product to more detail. In this process, which involves multiple technical disciplines, interfaces between components and technologies are assumed and components designed to meet these interface conditions. Some slack is introduced into the interface conditions to provide component designers with leeway so that redesign with altered interface values is not required very often and design can proceed rapidly to increasing detail.

In general, design is "largely completed" prior to the start of construction. A different organizational entity will then take the product of design, the plans and specifications, and carry out the physical construction from the model of the product developed during design. While the designer should understand and base many design decisions on the process of physical construction, this knowledge is seldom reflected in the construction contract documents. In fact, when one looks at typical U. S. contract drawings, one finds working drawings that lack greatly in detail. Much of the detail is left to shop or fabrication drawings developed by contractors and subcontractors who are responsible for actual physical interfacing while constructing in the field. Because this interfacing is done in the field, the construction process is slowed, prefabrication opportunities limited, rework rampant, and excess conservatism prevades design. Overcoming these problems is the goal of this research program.

Computer use is not new to the field of construction or its design disciplines. Analysis and component design programs are now commonly used in the majority of engineering offices for an increasingly wide variety of tasks. Commercial drafting systems are available for the production of contract drawings. While the majority of these CAD systems are limited to geometric information, more are becoming broader database driven representations of designs. As such, they are starting to carry more design information and will allow increasing integration of the design process.

Similarly, construction companies have been using computers for tasks such as construction scheduling, estimating, and financial management. CAD and design software is not widely used by contractors even though they are responsible for the production of numerous drawings.

In practice, the contractor does not have access to the decision making that took place during the design process, including the reasoning behind the setting of interface conditions between components and the designs of the components themselves. Such information would assist in construction planning and adaptation of the design to unanticipated field conditions or the like. It is as if the whole design process were incapsulated into the contract documents, which are then thrown over a barrier speed between design and construction, and the contractor left to infer the designers' thoughts from the meager tracings found in the representation of the product.

The construction industry is aware of current computer technology. It is continually expanding its analysis and design software, basing more and more of its software on database techniques, and even starting to apply simple rule-based expert systems techniques to the development of component design problems. But overcoming the problems shown above requires more than better use of existing computer methods. What is needed is a very different and superior "computer integrated design" system that integrates the whole process of producing the product.

#### Rationale for Goal Components

1. Less error in design - While designer are professional striving to produce error-free work, the scope and size of modern projects make this goal difficult to achieve. The quantity of design information and the number of people and technologies involved make coordination difficult. The computer can assist here by providing communications for interface assumptions, design requirements, and ongoing design results. At the same time it can provide a constant checking mechanism to monitor for inconsistencies, violations of interface assumptions, and, by knowing how components were designed, might even automate component redesign when errors are detected. Current systems are not organized to provide such facilities. Their information is limited to a representation of the product and do not capture design process information.
2. More detailed design - The reason that construction detailing is left to the field is that it requires understanding of the construction process, its tools and materials, as well as a clear understanding of how all components of the design interact. No one person in the design process has all of this information available at present. But a computer system, using some of the techniques required for design coordination, could provide and use this information. The consequences of having more detailed design would be to allow both more automation and more prefabrication because less field decision making would be required. Both would improve the cost effectiveness of the process and product.
3. Automation - Large scale automation is expected to improve the productivity of construction. While work is proceeding on the development of automation devices and techniques, parallel work is needed to provide information from design for the planning, management and control of the automation devices. While current design systems are organized to develop a representation of the final construction product, management of the automation devices

requires a representation of the continually changing construction environment, changing through the actions of both humans and robots as construction proceeds. Information needs for their management and methods for its generation must be developed. In addition, the characteristics of the product to allow and promote automation are likely to change. This will also lead to changes in the design process and design information.

4. Better planning - With present discrete information systems, planning involves manual interpretation and manipulation of results from multiple design and construction information systems, creating in the process yet another discrete set of data. As a result, planning is minimized, with professionals relying on experience and attendant routinized procedures. Undertaking better planning requires several new tools, including integrated access to a much wider variety of project information, a new generation of project planning tools that take project knowledge and generate and update plans, and plan management tools for coordinating the generation and use of project plans. This goal fits closely with those above.

5. Control systems - Control is the process of using previously generated plans, measuring actual outcomes for project development processes, analyzing variances between the two, and making decisions on requisite changes in plans or expectations. The problems stated for the previous goal are equally true for current control systems. Lack of good control systems constrains automation and more detailed design and limits our ability to recognize planning, design and construction errors. Good automated control systems should infer project status from the broad range of information integrated in the system and perform analysis and impact mitigation actions in the background during continuing project development.

6. Constructibility - Currently little is done to assure the constructibility of a design. Each of the designers in their discipline tries to design envisioning the construction technique and equipment to be used and having atleast some understanding of the characteristics introduced into the design by others. The integrated project information base, however, does not exist for verifying or automating this constructibility checking. Tools for doing so are therefore one of the goals.

Basic Research Topics Needed to Achieve Goals

The construction industry is currently making extensive use of computers in both design and construction processes. CAD, database management, and a variety of component design techniques, including expert systems, are widely used. The industry seems ready, willing and able to continue to expand such use. Such work is therefore not basic research.

But what the industry cannot do is to integrate its computational environment and thereby change how its business is carried out. Such integration required the development, testing and refinement of significant new technology which will be outlined here. The next section will then propose a research program for getting there.

Six major research thrusts, three basic technologies, three applications foci, have been identified. These are:

1. Object oriented DBMS for design and construction
2. A coordination blackboard for manipulating such information
3. Plan generation frames for plan management
4. Mechanisms for capturing and using component design heuristics (Application of explanation based learning.)
5. Construction process simulation
6. Robot management information

Each of these will be explained along with their interactions.

1. Object Oriented DBMS - The typical project deals with massive amounts of data. For broad use of these data, they are being organized and managed with database management systems so that numerous design and construction processes can access and update common data. The problem is that such databases generally contain only data about the product, ignoring process information about how, why, by whom, etc. They are therefore very bland, static, and able to respond only to requests for data.

Object oriented programming is a technique for knowledge representation coming out of the AI field. This technique utilizes the lack of need for differentiating between data and procedures. This mixing allows us to combine in any conceptual object its current data as well as procedures used for its generation and knowledge useful for other procedures that may operate on it. An object might know what other work must be completed before its plan generation capabilities should be utilized to produce more detailed design. Similarly, when design results are available, it would know what other object should be notified and what information sent to them.

Research involves the design of a new type of DBMS that incorporates such broader knowledge. It also involves determining what is the nature of such broader knowledge and how it should be represented. It also includes the development of demon processes to monitor DBMS manipulation and automate consistency checking and conflict resolution. Such work should be undertaken using realistic project activities such as construction schedule generation, estimate and budget generation from project descriptions and CAD output, etc. This work is the foundation for all other tasks.

2. Designer's blackboard - Multidiscipline design involves a diverse set of tasks, many of which are knowledge intensive. Typical CAD programs, available for individual tasks, are normally developed by different people and hardly communicate with each other. The designer acts as a communication medium between these tools, which is a laborious process. Further, several designers may be working on different aspects of the design problem. Hence, there is a need to develop a design management system that supports the controlled sharing of design data, while avoiding potential conflicts between the designers.

The purpose of the blackboard is to provide an environment that can efficiently handle heterogeneous sources of knowledge. This environment will provide a methodology for developing interfaces between various CAD tools. A mechanism, that will utilize concepts from truth maintenance systems, for avoiding conflicts between various design alternatives will be implemented as a part of this environment.

3. Plan generation frames - Representing knowledge in terms of rules has proven reasonably satisfactory for diagnostic type problems. However, a problem arises in solving design or planning type problems. Generating an initial design requires one to first define alternative solutions based on both fundamental physical laws and heuristics, then evaluate these solutions, and finally select the most appropriate one. Knowledge for such problems takes a procedural form that often requires iteration multiple trials, appropriately coached with a plan based approach.

Planning is the act of designing a set of actions (planning) or objects (design) that satisfy a given goal, before actually performing the actions or constructing the objects. A basic planning system would include a representation for the planned product (see above), such as a building, as well as abstract networks describing predefined plans. Such predefined plans can be represented in frames containing information about the goals they can

achieve, the constraints they impose, as well as any side effects that they add to the final product or process. To solve a problem the system is initially provided with a set of goals and user specified constraints. The system would then augment the constraints with its own domain specific laws. At this stage, the system's collection of plans might be searched in an attempt to satisfy the goals within the given constraints. Parallel action planning, planning with resource allocation, and heirarchical planning might also be required.

Research on plan-based reasoning can be applied to conceptual design of typical constructed facilities and to their development and construction planning. How one characterizes a design for such planning is a critical knowledge engineering issue, as is plan management.

4. Component design heuristics - While many researchers and practitioners are currently generating small, special purpose expert systems for small processes and component designs, none are concerned with how such capabilities might fit into a computer integrated design system. The issue in such an integrated system, beyond the fairly simple problem of utilizing a wide variety of such small tools, is to create an interactive engineering environment where the computer can, through observation and query of the engineer during his use of the system, infer and store for later playback, communication, and use in redesign the basis for engineering decisions which are normal hidden in the simple data describing the product.

This problem will involve a combination of a variety of approaches, including explanation based learning, interfacing with what is called deep or fundamental knowledge about physical systems behavior, and prior knowledge retrieval for interactive directed query. Initial application might deal with drywall partition layout and design and lead to the generation of information needed in the next two thrusts, construction simulation and robot management.

5. Construction process simulation - Using planning methods, the general approach to the construction process can be developed and further detailed into a set of discrete tasks to be carried out by different parties. While this information is necessary, it is not insufficient for many detailed decisions. Equipment selection, detailed estimating and productivity assessment, and even feedback into design may require operating simulation of the construction operations. Such simulation requires a continuous representation of the construction process, equivalent to scene management in animation. To be

economical, information for such a simulation must come automatically from the design process and the plan development. Research is needed here on information content and structure for such integration and how such information is used in simulation.

6. Information for robot management - This information is similar to the simulation information covered above. The computer is needed in taking the design information and planning and scheduling the operations and motions of the robots. In particular, such planning includes sequencing robot motions constrained by its mobility and operating characteristics and fed with its construction goals (e.g., component layout). This planning must recognize how the robot is changing its environment as it proceeds with its work. The plan, finally, would be downloaded into the robot for reasonably autonomous operation.

#### Recommended Initial Project

The above are a series of long term research goals and areas. Given the constraints of personnel available to work on this research program and available funding, one can identify and assign specific research topics. All of these topics fall into the areas discussed above.

1. Design Environment - This project involves the implementation and testing of the blackboard architecture and its use in heirarchical design and design coordination. The test would verify knowledge representation schemes using the design of building interior components, including architectural layout (partitions, doors, finishes), HVAC, plumbing, lighting and electrical systems.

2. Detailed Construction Planning and Simulation - This project involves construction scheduling and goes from plan representation to component installation. In particular, this work would interface with the WALBOT project by generating robot control information for partition construction from design plans.

3. Object-Oriented and Intelligent Database Management - The practicality of the integrated system suggested here is highly dependent upon its ability to deal with and share large quantities of data as well as deep and heuristic knowledge. This requires the development and interfacing with the blackboard of an object oriented database. In addition, this database can be made intelligent by embedding into it knowledge of the influence and meaning of data and its changes on other data and goals for the use of the data. Several appropriate applications may be identified,

including project control where automated project status reporting might lead to better construction management decisions and transportation network maintenance, where a board set of operating and condition data could be used more effectively for dynamic maintenance decisions. Most of this effort should be on the basic set of tools, with the application used to prove the concepts of the system.

Further research projects will be identified as resources become available.

Prepared for the U.S. Army Construction Engineering  
Research Lab. Workshop on Expert Systems for Construction Scheduling

## **CALLISTO: An Intelligent System for**

## **Supporting Project Management**

**Steven F. Roth**

Intelligent Systems Laboratory  
Robotics Institute  
Carnegie-Mellon University  
Pittsburgh, Pennsylvania 15213

March 1987  
Champaign, Illinois

Copyright © 1987 CMU, Robotics Institute

# CALLISTO: An Intelligent System for Supporting Project Management

## 1. Overview of Problem

The goal of the CALLISTO<sup>1</sup> group has been to apply results of artificial intelligence research to support the project management process through modeling of project environments and managerial and analytical expertise. This has included developing methods for supporting the creation, updating, analysis, evaluation and reporting of project plans and schedules, supporting tracking and reaction to project events, and supporting various aspects of communication and negotiation among project managers (for details, see Sathi, Fox & Greenberg, Sathi, Morton & Roth). For the purpose of this brief review, the focus will be on one aspect of the management problem: project instability or changeability and some ways in which we have attempted to alleviate the managerial difficulties associated with it.

The area of application is the management of large engineering projects, whose function is to produce new computer prototypes. Because of uncertain technology, activity outcomes, and competition, large engineering projects are plagued by continuous change in goals, implementation plans, cost and progress estimates, resource and materials availability, and other aspects of the project environment which result in the need for constant schedule updating. There are many managerial tasks which are difficult in these projects because of schedule changeability. As an example, consider the difficulty associated with assessing project status.

It is necessary for managers to determine the current status of a project throughout its course. Numerous dependencies exist not only among activities and resources, but among the product components which are being designed and assembled. Managers must be aware of any changes in plans which might influence their progress or assumptions. *In early stages of the project, this may mean analyzing updated plans or schedules to identify significant changes and their consequences for activities, resources, and products for which they are responsible.* A similar need exists for assessing activity progress against schedules during the execution of a project.

For managers to be able to use schedules to be aware of changes in the project, schedules must be updated promptly and accurately and managers must be able to analyze them quickly and frequently. These tasks are difficult, however, because of the large number of activities involved (often thousands) and the large number of managers whose plans must be integrated across many departments and locations. Updating a project-wide schedule requires an enormous information-gathering task which is usually performed manually by an operations group and substantially after the impact of changes has already been felt.

Even if schedules could be updated and maintained promptly, analyzing weekly schedule changes to evaluate their significance for a manager's concerns can be tedious and extremely difficult when there are thousands of activities and resource dependencies. Managers are not likely to make use of schedules unless they can rapidly locate the information that is most relevant.

Another reason why it is difficult for managers to use schedules to analyze project events is that

---

<sup>1</sup>The ideas in this report are the result of collaboration among the members of the CALLISTO project: Joe Mattis, Xavier Mesnard, Arvind Sathi and Mark Fox. This work has been supported by Digital Equipment Corporation.

schedules typically contain very little knowledge of the project because they only represent temporal and precedence constraints. This is especially important for the task of analyzing a project after completion. Because of the enormous expense and long life span of these projects (4-5 years), it would be advantageous to apply knowledge acquired during each project to subsequent ones. Knowledge acquisition and application is impeded by the high turnover of managerial staff even within a single project. Few managers play the same role in two consecutive projects. In order to facilitate transfer of expertise, it would be necessary to maintain a detailed description and history of the project events, decisions and activity outcomes. Accurate histories are also necessary to understand details and rationale of design decisions which result in numerous versions of the prototype that the project produced. CPM and PERT models do not convey much of the needed history.

As a result, updated schedules have not provided a realistic vehicle for communicating or analyzing change because they are not updated promptly, they do not contain sufficiently rich representations of projects, and the amount of information to be searched and analyzed to find relevant facts is prohibitive. Several research areas within the CALLISTO project have addressed these issues and are reviewed next.

## **2. CALLISTO Approaches**

### **2.1. Development of a Semantic Representation of Projects**

The majority of our initial work and much of our ongoing work has dealt with knowledge engineering and representation. Based on the previous success of the ISIS factory scheduling system, a schema-based (frame) representation of projects was developed using SRL (which has become the commercial product KnowledgeCraft). The goal was to develop a rich enough representation to support a variety of scheduling, analysis, and reasoning capabilities, as well as a detailed historical record of a project. As a result, the CALLISTO architecture separates declarative knowledge of prototypical concepts and project facts from expertise encoded in rule-based and procedural components.

CALLISTO's declarative representation includes:

- General epistemological concepts
  - time
  - causality
  - abstraction and aggregation
  - possession
  - change
- Domain concepts
  - organizational concepts and relationships (e.g. activity responsibility, departmental ownership of resources)
  - definition, classification, aggregation and abstraction of prototypical activities and resources (e.g. for assisting plan creation and evaluation, as well as analysis and scheduling at multiple levels of abstraction)
  - change in product configuration (e.g. representing phases and results of the engineering change order process, including relationships among part versions; relations between parts produced and people and activities which produce them)

- constraints on project activities and resources (e.g. a language for expressing and integrating flexible constraints on start criteria for activities, including resources and materials needed, alternative precedence requirements, interruptibility)
- representation of negotiation process (e.g. protocols for communicating about and establishing commitments on activity deliverables and dates)

Tests of the completeness of the representation have consisted of attempting to record complexities of project plans described during review meetings, representing a phase of a large engineering project (at Digital Equipment Co), and through development of functionalities for supporting and evaluating planning, for activity and resource scheduling and chronicling, as well as analysis and explanation of scheduling changes.

## 2.2. Interface Capabilities: automatic generation of text and graphical explanations of change

Our goal is to develop an approach to explanation for assisting managers in the analysis and search for relevant information across large updated schedules. By "explanation", we mean the analysis, interpretation, clarification, reporting and illustration of plans, schedule information, and conclusions produced by project management systems. Our focus is the identification and explanation of **change** in project schedules and databases.

The need for such a mechanism is apparent both in the changeable engineering environment which we have studied as well as in current commercial project management software. One trend in this software seems to be to provide managers with the ability to create and store numerous schedules representing different assumptions for "what-if" analyses, different schedule updates, and records of actual progress. Despite the growing ability to maintain numerous schedule versions (as well as increasingly richer representations), there has been very little work on methods to assist users in the **comparison** and **analysis** of these.

Current approaches to explanation in AI occur primarily in rule-based expert systems where the method of explanation is to present a modified trace of the inferences which led to some conclusion. This approach has little relevance to the problem of change explanation because the task is not only to understand how a schedule date (or project cost) was derived, but how it and many related variables changed from one situation to another.

Our approach to explanation extends a technique called *comparative analysis* (Kosy & Wise, 1984) that has been used in the explanation of change in a company's financial models. Based only on knowledge of the set of equations that relate variables in a spreadsheet and two sets of data (e.g. expected vs actual costs; costs across time periods), the system is capable of explaining the change in the value of any variable by analyzing the contribution of change in each variable from which it is derived. The system can answer questions like *Why did overhead expenses go up from 1985 to 1986?* and *Why did maintenance costs go up by \$30,000 even though electrical-repairs decreased by \$10,000?*

The first stage of our work extended this approach to schedule date explanation by explicitly representing the algebraic relationships underlying CPM and resource-scheduling. This provided the ability to answer questions like, *Why is the end-date of the schedule (or activity X) much later in the new version?* and *What effect did the increase in the duration of the CPU-DEBUG activity have on the end-date of the MILESTONE-1?* A sample answer might be: *The schedule end-date was later because of*

*changes in the durations of activities A, B, C, ....and H, which delayed the path from X through the end of the schedule. Only half of the changes affected the end-date of the schedule because of 20 days of slack in SCHEDULE-1 after activity C. Note changes in secondary paths converging at Q which are almost critical in schedule-2.*

Although previous work was limited to explaining change quantitatively, the next stage involved change identification and interpretation at many levels of understanding, depending on the depth required by the user or the knowledge that is available to the system. These levels include identifying the qualitative properties of activities, resources and other project entities and the ways they can be classified, aggregated, abstracted, and summarized in order to suggest directions for understanding the "reasons" for the changes and not just the quantitative mechanisms. As a result, the strictly quantitative answer above can be augmented with: .... *Most of the duration increases were due to activities of type DEBUGGING and were in the CPU-DESIGN department.*

This level of explanation relies on only a weak causal model of lateness and uses heuristics for grouping activities which were responsible for the change in the date in question. Managers can also direct the system to use any set of relationships in the knowledge base to breakdown causes of date changes or project-wide costs by asking questions like: *What role did activities which were the responsibility of DEPARTMENT-1 (alternatively, depend on CAD-MACHINES; part of DESIGN-PHASE-1) play in delaying activity X?* The answer is not only a single number, but also a breakdown of each category in terms of smaller activity groupings (e.g. further breakdowns of a department into sub-departments, resource classes like CAD-MACHINE into subclasses, schedule periods into smaller ones, workbreakdown hierarchies into more detailed activities, etc).

Our next efforts are to explain the "reasons" for changes or methods by which changes were produced. This may mean providing the rationale for changes when the system produces them (e.g. in automatic recalculation of durations based on evidence of changes in the project environment). It may also mean referring to other databases which track the process of negotiation and commitments among project managers which underly the activity schedule changes.

Finally, it has become clear that explanations cannot occur in natural language alone. There are many relationships which must be communicated with graphics. To provide this capability, we have begun a project called AUTOGRAPH, which is a system for automatically selecting and generating appropriate displays for illustrating information that needs to be conveyed to users. Using a library of styles that are appropriate for each domain (e.g. various styles of PERT, GANTT, resource profiles, hierarchical breakdowns, etc) and a description of the information needed to be conveyed by the explanation system, AUTOGRAPH selects and constructs an appropriate display. The display can then serve as an interface by which the user can peruse the project database from a particular perspective, request subsequent explanations or perform various editing operations. As a result, explanations occur as combinations of text and graphics and provide an interesting test-bed for studying the coordination of these two modes of communication.

An intelligent graphical agent such as AUTOGRAPH is necessary because the decision-making process for choosing an appropriate style is complex. Often users are unfamiliar with all the display styles available in a domain or system, unfamiliar with the criteria for choosing among styles for particular goals, or unfamiliar with a system interface and how to select, change and tailor styles to meet their goals.

Equally important is the fact that the same information-seeking goal (e.g. finding the causes of change in an activity's end-date) requires very different pictures because of differences in the nature of the information that is retrieved by the explanation system. It is often impossible to anticipate the best graphical style to peruse a knowledge base to answer a question. For example, understanding schedule changes for one activity in a schedule may best occur with a PERT diagram because the reasons are duration overruns in prior activities. Delays in another activity may best be understood using a resource profile for a small set of the project resources over a three week interval (because of resource bottlenecks).

The important point is that the answer to the question dictates the picture style and content and not the goal of the question. It could take considerable user effort to analyze the schedule in different styles until the most effective picture is discovered. Automatic selection of the appropriate picture may also expedite the next stage, which is finding a solution to an ongoing problem (e.g examining the resource profiles for the same activities over the next milestone).

In summary, our explanation and graphics system research is one method by which we can reduce the burden of identifying and analyzing changes across versions of large schedules. Our goal is to develop a system which is not restricted to CALLISTO conventions and methods, but is capable, with some interface, to explain changes in other project management systems.

### **2.3. Developing a distributed approach to project management systems**

The work on knowledge representation and explanation addressed the problem of adequately describing the project environment and easily finding relevant data from countless changes. The next area addressed the problem of *eliminating or reducing the feedback loop occurring between updates of schedules*. This loop is caused by the need for a central operations group which gathers information and assembles and disseminates project schedules. As pointed out earlier, for large projects whose managers are located throughout the country, this is a time-consuming task which reduces its utility.

Our approach has been to find ways to automate the acquisition and dissemination of schedule information. An advantage of this domain is that nearly everyone accesses computer terminals regularly and all systems are networked. As a result, we have been able to work on a continuum of methods for reducing delays in the update process and consequently in managers' reactions to project changes. They can be quickly characterized by the following scenarios in which a centralized operations group (COG) plays a progressively smaller role:

- 1 the COG gathers information, assembles schedules and sends reports
- 2 the COG gathers information and assembles schedules, but all managers have access to all aspects of schedules and must search these using explanation and query system
- 3 the COG gathers information and assembles schedules, but CALLISTO determines which changes are relevant to each manager and sends appropriate tailored reports when necessary. each manager can also construct a profile which communicates the kinds and level of detail of changes of interest
- 4 CALLISTO assumes the information gathering role. Managers use protocols for requesting updates. CALLISTO, with the collaboration of the COG, notifies managers who must share the responsibility for the proposed changes, updates the schedule, and performs the dissemination function described in 3. CALLISTO also requests information from managers when information is missing

This sequence of methods represents our previous approach of adhering to the centralized view of project management support. It has become apparent that this approach is insufficient. Another new area of research attempts to develop an alternative perspective of project management support from the one which is pervasive in classical scheduling approaches (CPM) and AI planning and expert systems. These approaches assume that project management is a centralized, hierarchical task, in which an expert applies knowledge to a problem description to decompose it to simpler problems which ultimately can be solved with known operations.

There are many contexts in which this is an appropriate view and it has been our approach for the first several years of CALLISTO research. In large computer engineering projects (and probably in large construction companies responsible for both design and construction), the planning process is more a combination of competitive and cooperative processes among many experts with different goals and functions within the company. As a result, the planning process is not a hierarchical decomposition of a large problem which is collectively solved by many experts. Instead it is a process of negotiation among agents governed by conflicting constraints, who strive to make commitments that enable the activities of a project to occur. It is no longer appropriate to think of an individual manager's schedule as a small portion of a project's schedule, since a manager may have responsibilities across several projects and therefore, conflicting goals.

As a result of this perspective we have begun to investigate ways to help manage the communication process either by providing a language for managers to communicate about project plans and conflicting constraints, or by providing methods by which some of the negotiation can be automated. Ultimately, our goal would be to represent the separate goals and constraints of each manager so that an agent maintaining a centralized view of a project (i.e. the CALLISTO system) can automatically communicate with and negotiate with agents representing the individual views of each manager.

# Research on the Use of Artificial Intelligence Techniques to Support Project Management

*Raymond E. Levitt  
Associate Professor of Civil Engineering  
Construction Engineering and Management Program  
Stanford University*

*Presentation by Catherine Perman  
PhD Candidate  
Department of Civil Engineering*

Our research in this area to date has focused on the development of a philosophy for the use of Artificial Intelligence (AI) techniques as aids in engineering project management.

We started by classifying the subtasks associated with project management as a taxonomy of separate functions (objective-setting, planning, scheduling and control) and levels of management (executive, work package and task). We then assessed the cognitive requirements for each project management subtask. [See **Figure 1** for a house-building illustration of the taxonomy].

Recognizing the cognitive requirements of each subtask and the limitations of existing computer tools for project management decision support, we have developed a set of guidelines for using AI and procedural programming techniques to support decision making in each phase and at each level of project management.

First, we propose that traditional domain-independent, "means-end" planners, may be valuable aids for planning detailed subtasks on projects, but that domain-specific planning tools are needed for work package or executive level project planning. Next, we propose that hybrid computer systems, using knowledge processing techniques in conjunction with procedural techniques such as decision analysis and network-based scheduling, can provide valuable new kinds of decision support for project objective-setting and project control, respectively. Finally we suggest that knowledge-based interactive graphics, developed for providing graphical explanations and user control in advanced knowledge processing environments, can provide powerful new kinds of decision support for project management. [These recommendations are summarized in **Figure 2**.]

The first claim is supported by a review and analysis of previous work in the area of automated AI planning techniques that we conducted over the last year. Our experience with **PLATFORM I, II and III**, a series of prototype AI-leveraged project management systems built between 1985 and the present, using the IntelliCorp *Knowledge Engineering Environment (KEE<sup>TM</sup>)*, provides the justification for the latter two claims.

The **PLATFORM** systems are a series of prototype hybrid AI/Procedural systems that were used to test out our notions about the value of AI in the domain of project management. While we continue to develop the ideas in these systems, concepts from the work have been implemented in a series of commercial grade systems for factory automation.

Current research is focusing on the use of planning systems that combine general search procedures such as *means-end* with domain specific knowledge implemented in frames and rules. We are in the process of testing and extending SIPE, a planner developed by David Wilkins of SRI, for executive level construction planning problems, and building extensions to PLATFORM I in KEE for project monitoring and knowledge-based schedule updating..

Research to date has been funded by a sabbatical leave grant from *IntelliCorp* and by seed funding through the *Stanford Construction Institute*.

This work is described more fully in the following papers:

Levitt, R.E., and Kunz, J.C., "Using Artificial Intelligence Techniques to Support Project Management," *Working Paper No. 1, Center for Integrated Facilities Engineering*, Departments of Civil Engineering and Computer Science, Stanford University, Stanford, CA, 1987.

Levitt, R.E., and Kunz, J.C., "Using Knowledge of Construction and Project Management for Automated Schedule Updating," *Project Management Journal*, December 1985.

Kunz, J.C., Bonura, T., Stelzner, M., and Levitt, R., "Contingent Analysis for Project Management Using Multiple Worlds," *Proceedings of First International Conference on Applications of AI in Engineering Problems*, Springer Verlag: 1986.

Objective Setting	Planning	Scheduling	Control
<b>Build House</b> <ul style="list-style-type: none"> <li>• choose location</li> <li>• total area, # BR, # BA</li> <li>• style, consider additional structures (garage, swimming pool)</li> <li>• budget</li> <li>• overall duration</li> </ul>		<ul style="list-style-type: none"> <li>• estimate the overall duration and cost using parameter estimates based on similar, past projects with adjustment</li> </ul>	<ul style="list-style-type: none"> <li>• monitor completion date, total cost, quality control vs. plans, schedules, and objectives</li> </ul>
<b>Work Package Level</b> <ul style="list-style-type: none"> <li>• choose materials (quality vs cost, time)</li> <li>• select construction methods (time, cost, quality, environmental impact)</li> </ul>		<ul style="list-style-type: none"> <li>• estimate activities durations and resource requirements based on crew productivity, equipment spreads, and/or subcontractors quotes</li> </ul>	<ul style="list-style-type: none"> <li>• monitor productivity and production rate data for work package (geotechnical data, concrete cylinder tests, etc)</li> </ul>
<b>Task Level</b> <ul style="list-style-type: none"> <li>• choose type of foundation (allowable settlement, cost, time)</li> <li>• construction method (crew size, equipment, cost, time)</li> </ul>		<ul style="list-style-type: none"> <li>• estimate by analogy with past projects data and/or by modeling the crews and equipment spreads</li> </ul>	<ul style="list-style-type: none"> <li>• monitor daily labor, material, equipment costs vs. schedules and budgets</li> <li>• monitor quality in detail (placement of rebar, concrete finishes, ...etc)</li> </ul>

Figure 1. A Taxonomy of Project Management Tasks for House Construction

	Objective-Setting	Planning	Scheduling	Control
Executive Level	Decision Analysis or Spreadsheet Algorithms, with ATMS and Multiple Worlds/Contexts	Interactive Graphics with Automated Planning Techniques: Domain-Independent	Interactive Graphics with Network-Based Scheduling Algorithms in a Knowledge-Processing Environment	Intelligent Database Access and Analysis Tools, with Model-Based Reasoning for Aggregating across Levels, Rescheduling, Replanning or Revising Objectives
Work Package Level	Network-Based ----- Heuristic Treatment of Time Computations			
Task Level			Crew Models ----- Interpretation of Past Data	

**Figure 2. Guidelines for Using AI and Procedural Techniques for Project Management Decision Support**

## CONSTRUCTION SCHEDULING: EXPERT SYSTEMS DEVELOPMENT

by Romey Ross

### The Crisis

A current report from the National Research Council calls for significant federal involvement in construction oriented Research and Development. The construction industry invests less in R & D than does any other major industry. Perhaps more importantly, it invests much less than many foreign construction industries. This wouldn't be a problem except for the fact that construction productivity has been stagnant for at least two decades. Coupled with this is the unnerving ascendancy of Japanese, Korean, and other international construction groups. Just like Detroit car makers we are facing a crisis of our own making. The ultimate issue in all this is how to improve productivity. Construction is raw - real world. It is typically conducted in a non-controlled environment where changes are common and surprises frequent. Anyone who has carried tools professionally can tell you to expect only a modest increase at the worker level. After all, human beings have certain physical limitations--people can move only so fast and carry only so much. Most construction tradesmen work fast and hard if they have: a) the right materials to work with; b) some idea of what to do next; c) coordination with others; and d) qualitative and quantitative feedback. The common theme here is productivity improvement depends on increasing the effectiveness of management at all levels.

Management in construction is typically of a type known as Project Management.

Project: an undertaking which may be unique, has special constraints (time and/or resources), and which is typically complex.

Management: Planning, Monitoring, Correcting and back to Planning.

Hence, Project-Management is a discipline built on Planning, Monitoring, and Correcting--in other words, Scheduling. Project Management must always emphasize the Value-Added or Payout-Ratio of its actions. There must always be an effort to be concise, to streamline, to simplify, and to distill. Further, management must communicate expectations, feedback of results, awareness of the Value-Added to the process, and the connections (chronological & other) as changes occur. Finally, management must engender belief in the plan. Tradesmen must "buy-in" to the plan (schedule) before monitoring and correcting will work.

### Why Expert Systems?

Efforts to improve management have begun to focus on the potential use of expert systems. Expert systems are a branch of the Artificial Intelligence tree (no contradiction of terms intended) which strive to "mimic the problem-solving and decision-making thought processes of human experts". In short, the goal is to have machines help non-experts function as effectively as experts. Our experiences in

implementing computer systems (since the mid-60's) have generated strong opinions regarding the importance of various factors in the success or failure of computer applications. Recent studies have indicated that human beings are almost uniformly motivated by the same things: recognition, respect, solicitation of input, the perception that coworkers take pride in their work, the opportunity to do a "good job", and money. Disincentives are equally consistent: cleanup of someone's mess, redoing almost any task, lack of recognition (or worse, punishment), isolation from feedback, or circumstances which prevent the performance of good work.

The foregoing observations shed light on software successes and failures we have experienced. Successful software reinforces the motivations and minimizes the disincentives. The reverse is true in most cases of failing software. Exploring possible expert systems, our most critical questions revolve around human engineering. Bitter experience indicates good designs on paper can be complete flops in the world of users. This is probably even more important with an "Expert" system which users may perceive as a threat to their jobs. This class of issues (human factors/engineering) comprises perhaps the most important set of system design concepts. Human engineering cannot be simply a band-aid type solution, although it frequently is. Rather, human factors must be designed from the very beginning.

What are some of these human factors? First of all, a system must not be condescending nor patronizing to the user. Most users of expert systems will not be novices in the area of automation. They are known as "transfer users." They can learn function keys and syntax rapidly because they are transferring knowledge from previous software experience. Consequently, complexity should not be wasted on babysitting users. Just as important, however, is the need for consistency of syntax throughout the program, and use of nonsensitive syntax to allow flexible phrasing and response. Furthermore, the system should have limited complexity, especially regarding help or special features. On-line help is much less important to a transfer user than good, solid, concise documentation. The system should also exhibit limited unique functions so the user is not overwhelmed by too many bells and whistles. Likewise, there is a need for mnemonic commands, logical progressions and nesting of the various system interface levels (good examples of this can be found in Lotus 123 and AMS Time Machine).

Human factors must be religiously incorporated in these seven steps to successful programming:

- 1) Plan
- 2) Emphasize a good user interface
- 3) Give the user what he wants
- 4) Make everything modular
- 5) Lock the final design
- 6) Document concisely and well
- 7) Test, test, test

Simple to say but difficult to achieve. The bottom line is: a system must be a help, not a hindrance.

## Features and Pitfalls

Effective scheduling expert systems fall into two classes:

- 1) A "front end generator" to feed an existing full featured scheduling program.
- 2) A "net tester" to analyze existing networks for "reasonableness"; to balance activity durations; to test completeness of activity lists in sub-nets; to confirm the presence/absence of appropriate design and procurement "hooks"; etc.

Ideally, these two primary systems should incorporate the following features:

- Low Cost
- Fast and Easy to Use
- Hardware lenient
- Distinct "Audit Trails"
- Believable, reasonable products
- Maximize productivity and effectiveness of Knowledge Workers (project support specialists). This implies integrated cost and schedule control systems.
- Output should be "quick and dirty" rather than "slow and perfect".
- Output should be graphic, based on exception reports and geared toward Visual Early Warning System layout.

## Conclusion

The preceding observations and assumptions add up to quite a tall order. Some of our most successful steps in the evolution toward artificial intelligence based systems have consisted of hardcopy Flowcharts and Checklists coupled with Procedures Manuals (12 at last count) written by company experts. Such steps are necessary precursors to interfacing Man and Machine. Like the Chinese symbols for crisis--one means opportunity, the other means danger--we see a future rich in risk and opportunity. We want to emphasize the Machine's role as rationalist, linear, logical, supportive partner, while maximizing the Human roles of intuition, multi-factor processing, and holism. Someday we may even see expert systems training of Scheduling or Project Management via interactive gaming, similar to the adult game *Interlude*. The future should be interesting.

**Construction Scheduling Issues:  
A Construction Firm's Perspective**

**Prepared for the USA-CERL Workshop  
on Expert Systems for Construction Scheduling**

**March 23-24, 1987  
Champaign, Illinois**

**by  
Paul M. Teicholz  
Guy F. Atkinson Company of California**

**INITIAL NETWORK CREATION**

**(1) NETWORK LOGIC**

An E.S. could help select typical subnets for selected types of work. There are typical sequences of activities that are normally required for given field operations, e.g., form, place, cure, strip for concrete work. These sequences could be generated from a CAD 3D model of a structure combined with an E.S. that contained knowledge of how a given type of structure was built.

**(2) NETWORK REASONABLENESS**

The initial network could be subjected to a number of tests for consistency and reasonableness, e.g.,

- (a) Are outdoor activities using a calendar with appropriate weather days?
- (b) Are indoor and other non-weather sensitive activities using a calendar without weather days?
- (c) Do comparable activities have reasonably similar duration or production rates?
- (d) Do the resources assigned to activities seem reasonable, based on the type of activity and quantity of work (based on comparison to estimating standards)?

**UPDATING OF NETWORK**

**(1) NETWORK LOGIC**

Is the network logic being followed in the field (as indicated by actual start and finish dates)? If not, i.e., activities are being started out of sequence, this situation needs to be flagged, so that the proper start dates can be entered on the network logic revised. An E.S. is not needed for this.

(2) DURATION

An E.S. could compare the revised total duration of in process and unstarted activities to determine whether these are reasonable based on the duration of similar completed activities. For example, if the completed steel erection activities of a 10-story building that is one-half completed show a 20% overrun (on the average), then the remaining durations for steel erection should be comparably increased.

(3) RESOURCES

If actual resources are being allocated to activities, then an E.S. could be used to analyze whether the resource levels are reasonable. Three comparisons are required:

- (a) Actual resource usage rates for comparable activities, e.g., work hours per day for concrete placing operation.
- (b) Actual resource usage rates vs. budget resource usage rates.
- (c) Actual resource usage rate for to-date duration vs. forecast remaining usage rate (for a given activity). If these are very different, either the remaining duration or remaining resources need to be changed.

(4) RISK ANALYSIS

Using Monte Carlo Analysis to calculate a probability distribution for meeting specified milestone dates (based on duration variability estimates derived from the type of work and to-date project experience), an E.S. could analyze the results and point out where changes in resource levels or network logic might be desirable to increase the probability of meeting due dates. This requires a very high level of sophistication, but is exactly the type of analysis that is often ignored because of insufficient understanding and/or lack of time. An E.S. might address both of these impediments.

USA-CERL DISTRIBUTION

Defense Technical Info. Center 22314

ATTN: DDA (2)

U.S. Govt Print Office 22304

Receiving Sect/Depository Copies (2)

END

10-87

DTIC